

## **VORLÄUFIGE DOKUMENTATION DES BEFEHLSSTATZES DES TVT CBKD COLOR CONTROLLERS**

Stand: 23. Juni 2005

Für eine aktuelle Version siehe: [www.tv-terminal.de](http://www.tv-terminal.de)

(c) Viebig Datensysteme GmbH 2005

### **Terminal Steuercodes**

Wenn ein ungültiges Kommando empfangen wird, ertönt ein Tonsignal. Wenn kein Tonsignal zu hören ist, bedeutet das, daß das Kommando den syntaktischen Erfordernissen des Terminals genügt, aber nicht implementiert ist.

# - ein- bis zweistellige dezimale Parameter, führende Nullen werden ignoriert

### **Empfangene Steuerzeichen**

Name	HexCode	Funktion
Null	\$00	Hat keine Aktion zur Folge. Kann ggf. als Füll-Zeichen (Übertragungspause) verwendet werden, wenn der Befehl die Löschung des gesamten Bildschirms beinhaltet (mindestens 20 Zeichen senden). Ansonsten sind bei den meisten Befehlen keine Wartezeiten notwendig.
Bell	\$07	Der TV-Terminal-Controller gibt kurzes Tonsignal an Pin15 (bzw. beim Evaluation Board über den Scart-Connector Pins1+3) aus.
Backspace	\$08	Verschiebt den Cursor um eine Position nach links. Wenn der linke Rand erreicht wird, stoppt der Cursor.
Horizontal-TAB	\$09	Verschiebt Cursor ein Zeichen vorwärts (wie SPACE)
Linefeed	\$0A	Erzeugt eine Linefeed- oder NewLine Operation je nach Konfiguration (siehe Linefeed/NewLine Mode)
Vertical Tab	\$0B	Wird behandelt wie Linefeed
Form feed	\$0C	Wird behandelt wie Linefeed
Carriage Return	\$0D	Transferiert den Cursor an den linken Rand der aktuellen Zeile. Wenn Auto-Linefeed aktiviert ist wird stattdessen an den linken Rand der nächsten Zeile gesprungen.
ESC (Escape)	\$1B	Escape signalisiert zusammen mit „[" den Beginn einer Steuer-Sequenz

### **Tastatur**

Bei festgehaltener CTRL-Taste und gedrückter Taste in A-Z werden die ASCII-Codes \$01-\$1A gesendet.

Bei Betätigung der Tasten F1-F6 sendet das TV-Terminal die Strings ESC[OP bis ESC[OU.

Bei Betätigung der Cursor-Tasten werden folgende Codes gesendet:

UP: ESC[A

DOWN: ESC[B

RIGHT: ESC[C

LEFT: ESC[D

Außerdem bewegt sich der Cursor auf dem Bildschirm (nicht wenn Remote Echo aktiv ist oder das Keyboard gesperrt ist).

Der Status der CAPS LOCK – Taste wird in der Statuszeile angezeigt

Durch wiederholtes Betätigen von CTRL-F10 bzw. SHIFT-CTRL-F10 können die vertikalen Positionen der unteren bzw. oberen Statuszeile an den jeweiligen Bildschirm angepasst werden. Ein an das Terminal angeschlossener Microcontroller (der Host) kann mit dem Befehl ESC[7n die Position der Statuszeilen abfragen, oder mit ESC[<upos>;<lpos>v diese Positionen neu setzen.

Mit ALT-F11 kann das Scart-Schaltspannungs-Signal an Pin3 des TVT-Controllers ein- und ausgeschaltet werden.

Betätigung von F12 löscht den Bildschirm.

### **RESET und Signalisierung des Terminal RESET an den Host**

Bei Betätigung des Reset-Buttons auf der Evaluationsplatine, bzw. wenn der Reset-Pin des TVT-Controllers auf LOW gezogen wird, erfolgt ein Kaltstart des Terminals. Hierbei wird das Datum auf den 1. Januar 2001 und die Uhrzeit auf 00:00:00 Uhr zurückgesetzt. Der zulässige setzbare Datumsbereich ist 1. Januar 2001 bis 31. Dezember 2098. Außerdem wird die Position der Statuszeilen am oberen und unteren Bildrand auf den Initwert zurückgesetzt, und ein Reset der angeschlossenen Tastatur durchgeführt.

Bei Betätigung von CTRL-ALT-DEL auf der an das Terminal angeschlossenen Tastatur, sowie wenn das Terminal die Sequenz ESC [ z empfängt, wird ein Warmstart des TVT-Controllers durchgeführt. Hierbei bleiben Uhrzeit und Datum, sowie die Positionen der Statuszeilen erhalten.

Bei jedem RESET (Kaltstart und Warmstart) wird außerdem Pin2 des Expansion-Connectors für ca. 200µs auf LOW gezogen. Hiermit kann einem angeschlossenen µC-System der RESET des TVT-Controllers angezeigt werden. Das µC-System kann dann z.B. den Bildschirm neu initialisieren und aufbauen, oder auch selbst einen RESET durchführen und anschliessend beim TV-Terminal einen Kaltstart durchführen. *WICHTIG: Das Signal ist bei Verwendung unbedingt extern mit einem Pull-up-Widerstand (z.B. 10kOhm) gegen +5V zu versehen !*

### **ESCAPE-Sequenzen**

- Alle nachfolgend gelisteten ESCAPE-Sequenzen zur Konfiguration des Terminals sind nur im Text-Mode wirksam, mit Ausnahme der Befehle ESC[=I (zurück in den Text-Mode schalten, kleines „L“) sowie ESC[z (RESET/Warmstart des Terminals), die auch im Balkengrafik-Modus Einfluß nehmen.
- Alle Zahlen-Parameter sind dezimal und ein- oder zweistellig anzugeben .
- Die Befehlseingabe über die Tastatur schlägt fehl, wenn die CAPS LOCK Taste aktiviert ist.
- Die VT100-kompatiblen Befehle sind mit einem Stern „\*“ gekennzeichnet.

### **Setup / Reset Commands**

<b>ESC[HH;MM;SS?s</b>	<b>Uhrzeit setzen HH[00-23];MM[00-59];SS[00-59]</b> Die Uhrzeit wird im 24-Stunden-Format eingegeben. Die gesetzte Uhrzeit wird bei einem Warmstart per CTRL-ALT-DEL nicht neu initialisiert.
<b>ESC[DD;MM;YY?t</b>	<b>Datum setzen DD[01-31];MM[01-12];YY[01-98]</b> Der Wochentag wird automatisch berechnet. Das angezeigte Datum wird bei einem Warmstart per CTRL-ALT-DEL nicht neu initialisiert. ANMERKUNG: Nach dem Befehl für das Setzen des Datums sollte ca. 500ms gewartet werden, bevor weitere Daten an das Terminal übermittelt werden, da sonst gesendete Daten verloren gehen können.
<b>ESC[2h</b>	<b>* Disable keyboard input</b> Keyboard ist komplett gesperrt. Dies wird durch ein Symbol in der Statuszeile angezeigt

<b>ESC[2l</b>	<p><b>* Enable keyboard input (DEFAULT)</b></p> <p>Rücknahme der Keyboard-Komplettsperre. Wenn auch die Keyboard-Konfigurationsbefehle mit dem Befehl ESC[&gt;82h gesperrt waren, müssen diese separat mit dem Befehl ESC[&gt;82l wieder entsperrt werden. Die beiden Sperren existieren unabhängig voneinander. Die Sperre mit der höchsten Priorität (= Komplettsperre) wird jeweils in der Kommandozeile angezeigt.</p>
<b>ESC[&gt;82h</b>	<p><b>Disable configuration by keyboard</b></p> <p>In diesem Modus können über das Keyboard keine ESC-Sequenzen an das Terminal übermittelt werden (das ist beispielsweise dann notwendig, wenn diese ESC-Sequenzen über die serielle Schnittstelle herausgesendet werden sollen). Des weiteren sind die Funktionstasten F10-F12 verriegelt und CTRL-ALT-DEL (Terminal Reset) ist gesperrt. Die Cursorposition kann im Local Mode allerdings noch mit den Pfeiltasten verändert werden, außerdem funktioniert Backspace. Soll auch dies gesperrt werden, ist der Remote Mode zu aktivieren (was ohnehin angeraten wird). Diese Sperre wird als Symbol in der Statuszeile angezeigt, sofern nicht das Keyboard komplett gesperrt ist (dann wird das Komplettsperre-Symbol dargestellt).</p>
<b>ESC[&gt;82l</b>	<p><b>Enable configuration by keyboard (DEFAULT)</b></p> <p>Rücknahme der Keyboard-Konfigurationssperre. Falls das Keyboard mit dem Befehl ESC[2h komplett gesperrt war, muß diese Sperre separat mit dem Befehl ESC[2l wieder entsperrt werden. Die beiden Sperren existieren unabhängig voneinander. Die Sperre mit der höchsten Priorität (= Komplettsperre) wird jeweils in der Kommandozeile angezeigt.</p>
<b>ESC[12h</b>	<p><b>* Send/Receive mode: Local echo (DEFAULT)</b></p> <p>Alle an der Tastatur eingegebenen sendbaren Zeichen werden nicht nur über die serielle Schnittstelle herausgesendet, sondern auch gleichzeitig auf dem Fernsehschirm angezeigt. Bei Anschluss an einen Host empfiehlt es sich, auf Remote Echo Mode umzuschalten, um die Tastatureingaben filtern zu können. In diesem Fall sollte auch die „Konfigurationsmöglichkeit per Keyboard“ abgeschaltet werden. Im Remote echo Mode können zwar von der Tastatur keine ESC-Befehle an das Terminal gesendet werden, aber die Funktionen der Tasten F10-F12 sowie der Reset per CTRL-ALT-DEL funktioniert weiterhin. Die kann mittels Übersendung von ESC [ &gt; 8 2 h unterbunden werden.</p>
<b>ESC[12l</b>	<p><b>* Send/Receive mode: Remote echo</b></p> <p>Alle an der Tastatur eingegebenen sendbaren Zeichen werden lediglich über die serielle Schnittstelle herausgesendet. Auf dem TV-Schirm erscheinen nur Zeichen, die über die serielle Schnittstelle empfangen werden.</p>
<b>ESC[20h</b>	<p><b>* Linefeed/NewLine Mode: select NewLine Mode (DEFAULT)</b></p> <p>Die RETURN-Taste sendet bei Betätigung CR LF (\$0D \$0A). Empfangene Linefeed (\$0A), Vertical Tab (\$0B) oder Formfeed (\$0C) bewegen den Cursor an den linken Rand der nächsten Zeile (fügen also ein CR==\$0D hinzu).</p>
<b>ESC[20l</b>	<p><b>* Linefeed/NewLine Mode: select Linefeed Mode</b></p> <p>Die RETURN-Taste sendet bei Betätigung nur CR (\$0D). Empfangene CR (\$0D) bewegen den Cursor an die äußerste linke Position der aktuellen Zeile.</p>

Anmerkung: Diese VT100-ESC-Sequenzen bieten leider manchmal zu wenig Flexibilität. Die ansonsten empfehlenswerte Freeware SUPERTERM (<http://www.mpfreezone.com>), ein sehr gutes schlankes Werkzeug für Microcontroller-Systementwicklung mit Windows PCs, kann leider bei Betätigung der RETURN-Taste nur CR (\$0D) und kein Linefeed senden. Mit RETURN läßt sich daher kein Zeilenwechsel erreichen. Abhilfe schaffen die hier eingeführten proprietären Befehle ESC[>8h/I sowie ESC[>8h/I (siehe unten).

<b>ESC[&gt;8h</b>	<b>Enable auto line feed on carriage return</b> In diesem Modus springt der Cursor bei einem empfangenen Carriage Return (\$0D) nicht nur zum linken Bildrand, sondern verhält sich so, als wäre \$0D \$0A (Carriage Return und Linefeed) empfangen worden, d.h. er springt zum linken Bildrand der nächsten Zeile.
<b>ESC[&gt;8l</b>	<b>Disable auto line feed on carriage return (DEFAULT)</b> Wird ein CR-Zeichen (Carriage Return, \$0D) vom TV-Terminal empfangen, dann springt der Cursor zum linken Bildrand.
<b>ESC[&gt;9h</b>	<b>Enable auto carriage return on line feed (DEFAULT)</b> In diesem Modus springt der Cursor bei einem empfangenen Carriage Return (\$0D) nicht nur zur nächsten Zeile bei unveränderter Spalte, sondern verhält sich so, als wäre \$0D \$0A (Carriage Return und Linefeed) empfangen worden, d.h. er springt zum linken Bildrand der nächsten Zeile.
<b>ESC[&gt;9l</b>	<b>Disable auto carriage return on line feed</b> Wird ein LF-Zeichen (Linefeed, \$0A) vom TV-Terminal empfangen, dann springt der Cursor zur nächsten Zeile bei unveränderter Spalte.

Anmerkung: Wenn diese Auto-Linefeed-Funktion mit ESC[>8h aktiviert wird, dann muß sie wieder deaktiviert werden, **bevor** ESC[20h/l angewendet wird, da ESC[20h/l keinen Einfluß auf Auto-Linefeed hat. Nur wenn dies beachtet wird, verhalten sich die VT100-Befehle ESC[20h/l Norm-konform.

<b>ESC[&gt;19h</b>	<b>BACKSPACE sendet BS-SPC-BS</b> Falls das Backspace-Zeichen von der Gegenstelle nur als „Cursor um eine Position nach links“ interpretiert wird, schafft dieser Befehl Abhilfe.
<b>ESC[&gt;19l</b>	<b>BACKSPACE sendet nur BS (DEFAULT)</b> So wie es eigentlich sein sollte.
<b>ESC[&gt;84h</b>	<b>Loopback-Modus ON</b> Alle über die serielle Schnittstelle empfangenen Zeichen werden über diese sofort wieder herausgesendet (Ausnahme: ESC-Sequenzen).
<b>ESC[&gt;84l</b>	<b>Loopback-Modus OFF (DEFAULT)</b> Die über die serielle Schnittstelle empfangenen Zeichen werden nicht wieder herausgesendet.
<b>ESC[?25h</b>	<b>Command Display in Status Line ON (DEFAULT)</b> Wenn das Terminal den Anfang einer Escape-Sequenz erkennt (ESC[ wurde empfangen), dann schaltet die Statuszeile von Datums-/Uhrzeitanzeige auf Befehlsanzeige um und stellt den Befehl dar. Nach Empfang des letzten Befehls-Zeichens wird die Statuszeile unverzüglich wieder zurückgeschaltet. Dieses Feature ist sehr hilfreich, man kann die meisten Terminal-Befehle direkt durch eintippen auf der Tastatur ausprobieren, und erkennt sofort, wenn man sich vertippt hat. In diesem Fall am besten ESC drücken (was den Befehl abbricht), da eine Korrektur falsch eingetippter Befehle mittels BACKSPACE nicht möglich ist.
<b>ESC[?25l</b>	<b>Command Display in Status Line OFF</b> Wenn die Befehle vom Host über die serielle Schnittstelle kommen, und das Flimmern stört (natürlich tut es das !), dann diese Anzeige mit ESC[?25h vorher abschalten.

**ESC[<elem>;<col>q Define colors for Textscreen-Elements (<elem> in 0..23 und <col> in 0..15)**

Der Text-Bildschirm besteht aus 24 Elementen (0..23):

- 0: erste obere Statusframe-Linie
- 1: zweite obere Statusframe-Linie
- 2-19: 18 Textzeilen
- 20: erste untere Statuszeile
- 21: Datum und Uhrzeit
- 22: Statusanzeigen (CAPS LOCK etc.)
- 23: zweite untere Statuszeile

Diesen Elementen können die folgenden Farben zugeordnet werden:

- 0: SCHWARZ
- 1: PINK
- 2: GRÜNTÜRKIS
- 3: ORANGE
- 4: GELBGRÜN
- 5: BLAUTÜRKIS
- 6: VIOLETT
- 7: TÜRKIS
- 8: GELB
- 9: HELLROT
- 10: HELLGELB
- 11: HELLGRÜN
- 12: HELLTÜRKIS
- 13: HELLBLAU
- 14: GRAU
- 15: WEISS

**ESC[7n Report Status Lines Position**

Gibt (ausschließlich) an der seriellen Schnittstelle die Position der Statuszeilen im Format ESC[<upos>;<lpos>V aus

**ESC[<upos>;<lpos>v Adjust Status Lines (<upos> und <lpos> in 0..15, DEFAULT==6)**

Die Position der oberen (<upos>) und unteren (<lpos>) Statuszeilen kann in jeweils 16 Stufen vertikal variiert, und so an den jeweiligen Bildschirm angepasst werden. Die gesetzte Position bleibt auch nach einem Warmstart (per CTRL-ALT-DEL) erhalten. Beispiel: ESC[8;10v (höherer Wert bewegt Statuszeilen in Richtung Bildmitte)

**ESC[?15h Disable Status Line**

Schaltet die Statuszeile ab.

**ESC[?15l Enable Status Line (DEFAULT)**

Schaltet die Statuszeile ein.

**ESC[?75h Disable SCART Switch**

Schaltet die Wechsellspannung an Pin 3 des TV-Terminal-Controllers ab, die zur Erzeugung der 12V-SCART-Schaltspannung verwendet werden kann (mittels Spannungswandlung wie beim TV-Terminal COLOR Evaluation Kit). Dies bewirkt die Zurückschaltung des TV-Gerätes auf den per Fernbedienung angewählten Eingang.

**ESC[?75l Enable SCART Switch**

Schaltet die Wechsellspannung an Pin 3 des TV-Terminal-Controllers ab, die zur Erzeugung der 12V-SCART-Schaltspannung verwendet werden kann (mittels Spannungswandlung wie beim TV-Terminal COLOR Evaluation Kit). Diese

Schaltspannung auf Pin8 der SCART-Schnittstelle bewirkt die vorübergehende Umschaltung des TV-Gerätes auf den SCART-Eingang, ohne daß manuell per Fernbedienung auf AV umgeschaltet werden muß.

**ESC[?16h**                      **Disable Display**  
Der Bildschirminhalt (Zeile 1-18) wird unsichtbar geschaltet.

**ESC[?16l**                      **Enable Display (DEFAULT)**  
Der Bildschirminhalt (Zeile 1-18) wird sichtbar geschaltet.

**ESC[z**                        **Reset Terminal to power-up configuration**  
Bewirkt einen Warmstart des TV-Terminals (auch im Barchart-Modus). Datum und Uhrzeit, sowie die Position der Statuszeilen vor dem Warmstart, bleiben erhalten.

### Cursor Operationen

**ESC[s**                        **\*Save Cursor**  
Speichert die aktuelle Cursorposition sowie den origin mode status (relativ/absolut), der mit ESC[?6h bzw ESC[?6l gesetzt wurde.

**ESC[u**                        **\*Restore Cursor**  
Stellt die gespeicherte Cursorposition und den gespeicherten origin mode status wieder her.

**ESC[#A**                      **\*Cursor Up**  
Verschiebt den Cursor um # Positionen nach oben, limitiert vom Bildschirmrand bzw. den Grenzen des scoll area im relativen origin mode. Wenn kein Parameter angegeben wird, ist das Argument == 1.

**ESC[#B**                      **\*Cursor Down**  
Verschiebt den Cursor um # Positionen nach unten, limitiert vom Bildschirmrand bzw. den Grenzen des scoll area im relativen origin mode. Wenn kein Parameter angegeben wird, ist das Argument == 1.

**ESC[#C**                      **\*Cursor Right**  
Verschiebt den Cursor um # Positionen nach rechts, limitiert vom rechten Bildschirmrand. Wenn kein Parameter angegeben wird, ist das Argument == 1.

**ESC[#D**                      **\*Cursor Left**  
Verschiebt den Cursor um # Positionen nach links, limitiert vom linken Bildschirmrand. Wenn kein Parameter angegeben wird, ist das Argument == 1.

**ESC[x;yf**                    **\*Horizontal and Vertical Position**  
**ESC[x;yH**                    **\*Cursor Position**  
Setzt den Cursor in Zeile x (Zeilennummer in 1..18) und Spalte y (Spaltennummer in 1..50). Ist der relative origin mode mit ESC[?6h aktiviert, dann kann der Cursor nur innerhalb des Scrollbereichs gesetzt werden (siehe auch ESC[?6h bzw. ESC[?6l). Die Ränder des Bildschirmbereichs bzw. ggf. die Grenzen des Scrollbereichs limitieren den setzbaren Bereich.

**ESC[>5h**                    **Cursor Invisible**  
Schaltet den Cursor unsichtbar

**ESC[>5l**                    **Cursor not Invisible (DEFAULT)**

Schaltet den Cursor sichtbar

**ESC[>11h**

**Cursor static**

Cursor blinkt nicht, ist ständig sichtbar (sofern nicht unsichtbar geschaltet)

**ESC[>11l**

**Cursor blinking (DEFAULT)**

Cursor blinkt (sofern sichtbar geschaltet)

**ESC[#J**

**\*Erase in Display**

Der Parameter darf die Werte 0..2 annehmen und bewirkt folgendes:

- 0: Das gesamte Display ab der aktuellen Cursorposition wird gelöscht.
- 1: Das gesamte Display bis zur aktuellen Cursorposition wird gelöscht
- 2: Das gesamte Display wird gelöscht

Bei allen drei Modi wird ebenfalls das Zeichen an der Cursorposition gelöscht. Der Cursor behält seine Position.

ANMERKUNG: Nach diesen Befehlen sollte kurz gewartet werden (ca. 40 ms sollten ausreichen) bis die Löschung beendet ist. Anderenfalls können möglicherweise im direkten Anschluss an diesen Befehl geschriebene Zeichen verlorengehen, da sie vom Löschvorgang erfaßt werden.

**ESC[#K**

**\*Erase In Line**

Der Parameter darf die Werte 0..2 annehmen und bewirkt folgendes:

- 0: Die gesamte Displayzeile wird ab der aktuellen Cursorposition gelöscht.
- 1: Die gesamte Displayzeile wird bis zur aktuellen Cursorposition gelöscht
- 2: Die gesamte Displayzeile wird gelöscht

Bei allen drei Modi wird ebenfalls das Zeichen an der Cursorposition gelöscht. Der Cursor behält seine Position.

**ESC[#P**

**Delete Characters on the right**

Löscht eine dem Parameter entsprechende Anzahl von Zeichen rechts vom Cursor einschließlich des Zeichens an der Cursorposition

**ESC[#M**

**Delete Characters on the left**

Löscht eine dem Parameter entsprechende Anzahl von Zeichen links vom Cursor einschließlich des Zeichens an der Cursorposition

**ESC[?7h**

**\*Auto wrap mode on (DEFAULT)**

Schaltet den automatischen Zeilenumbruch ein. Wenn sich der Cursor am rechten Bildschirmrand befindet, erscheinen die ankommenden Zeichen in der nachfolgenden Zeile. Der Bildschirm scrollt, wenn der Cursor das Ende des Scrollbereiches erreicht hat. Auto Wrap wird in der Statuszeile durch ein Symbol angezeigt.

**ESC[?7l**

**\*Auto wrap mode off**

Schaltet den automatischen Zeilenumbruch ab. Wenn sich der Cursor am rechten Bildschirmrand befindet, überschreiben die ankommenden Zeichen jeweils das letzte Zeichen in dieser Zeile und die Cursorposition bleibt unverändert.

**ESC[Pt;Pbr**

**\*Select scrolling region (DEFAULT ist ESC[P1;18r gesetzt)**

Pt ist der Parameter, der die „Top“-Zeile der Scrolling-Region definiert und muß im Bereich 1-18 liegen. Für den „Bottom“-Parameter Pb gilt derselbe Bereich, zusätzlich muß gelten: Pb>Pt. Dieser Befehl setzt den Cursor ebenfalls auf die

Home-Position. Abhängig vom origin-Mode ist das entweder in der linken oberen Ecke des Bildschirms oder am linken Rand der ersten Zeile des Scrollbereichs „Top“). Zusätzlich wird der Bildschirm komplett gelöscht.

<b>ESC[?6h</b>	<b>*Origin Mode relative – Select HOME Position in Scrolling Region</b> Setzt die HOME-Position des Cursors (für die Kommandos ESC[f und ESC[H) an den linken Rand der ersten Zeile der Scrolling Region. Die Zeilennummern als Parameter für die Befehle ESC[x,yf und ESC[x,yH starten jetzt mit der ersten Zeile (Zeile 1) des Scrollbereiches. Der Cursor kann den Scroll-Bereich nicht verlassen.
<b>ESC[?6l</b>	<b>*Origin Mode absolute – Select normal HOME Position (DEFAULT)</b> Setzt die HOME-Position des Cursors (für die Kommandos ESC[f und ESC[H) in die linke obere Ecke des Bildschirms (d.h. in Zeile 1) . Die Zeilennummerierung ist absolut(e), d.h. unabhängig vom Scroll-Bereich.
<b>ESC[S</b>	<b>Scroll Screen Content one Line up</b> Scrollt den Inhalt des scrollbaren Bildschirmbereiches um eine Zeile nach oben. Am unteren Rand des Scroll-Bereiches erscheint eine Leerzeile.
<b>ESC[T</b>	<b>Scroll Screen Content one Line down</b> Scrollt den Inhalt des scrollbaren Bildschirmbereiches um eine Zeile nach unten. Am oberen Rand des Scrollbereiches erscheint eine Leerzeile.
<b>ESC[=h</b>	<b>Switch to Barchart Graphics Mode</b>
<b>ESC[=l</b>	<b>Switch to Text Mode</b>

#### Data Request from Host

---

Nach allen Data Request Befehlen sollte generell ca. 40ms gewartet werden, bevor neue Daten oder Befehle an das Terminal gesendet werden. Am einfachsten geht das, indem 40 Null-Bytes (\$00) an das Terminal gesendet werden.

<b>ESC[1n</b>	<b>Request Time</b> Das Terminal sendet als Antwort ESC[hh:mm:ssS, also z.B. um 8:20 Uhr und 43 Sekunden: ESC[08;20;43S
<b>ESC[2n</b>	<b>Request Date</b> Das Terminal sendet als Antwort ESC[dd:mm;yyS, also z.B. am 24. Juni 2005: ESC[24;06;05T.
<b>ESC[6n</b>	<b>Report Cursor Position</b> Das Terminal sendet als Antwort ESC[xpos,yposR, (xpos in 1..18 und ypos in 1..50), also z.B. ESC[17;36R wenn sich der Cursor in Zeile 17 in der 36. Spalte befindet.



## ESC[7n

### Report Frame offsets

Mit dieser Funktion kann der Host die Position der Statuszeilen auf dem Bildschirm abfragen, deren Position der Benutzer selbst über das Keyboard mit CTRL-F10 bzw. SHIFT-CTRL-F10 beeinflussen kann. Es könnte also innerhalb der vom Host-Computer eine Konfigurationsfunktion für die Einstellung der Lage der Statuszeilen geben. Der Benutzer wird hier aufgefordert, die Statuszeilen korrekt zu justieren und das Ende seiner Operation durch Drücken der ENTER-Taste zu bestätigen. Nachdem der Benutzer die ENTER-Taste gedrückt hat, fragt der Host mit diesem Befehl die Position der Statuszeilen ab und speichert diese bei sich als Default-Werte ab.

Das Terminal sendet als Antwort ESC[upos;lposV (Parameter jeweils im Bereich 0..15), also erst die Position der oberen Statuszeile (upos, bestehend aus zwei Linien), und dann die Position der unteren Statuszeile (zwei Linien sowie Datum, Uhrzeit und Statusinformationen).

Der Host kann jederzeit (z.B. nachdem er einen Warmstart des Terminals über dessen Reset-Leitung durchgeführt hat, die Position der Statuszeilen mit dem Befehl ESC[upos;lposv verändern.

## Grafik-Modus

Der Grafikmodus ermöglicht die Ausgabe farbiger Balkendiagramme. Im Grafikmodus ist das Gebiet des 18-zeiligen Textbildschirmes folgendermaßen aufgeteilt:

In Zeile 1 (Kopfzeile) steht sinnvollerweise die Einheit der Y-Achse (z.B.: °C) und anschliessend die Überschrift der Grafik. Die volle Zeilenbreite von 50 Zeichen ist verfügbar.

Zeile 2-17, also 16 Zeilen belegt der eigentliche Grafikbereich.

In Zeile 18 schließlich steht die Beschriftung der X-Achse (incl. der Einheit).

Bis zu 72 Balken können im Grafikbereich gleichzeitig dargestellt werden. Jeder Balken des Diagrammes ist aus bis zu 90 kleinen „Klötzchen“ aufgebaut und kann eine von insgesamt acht Farben annehmen:

0 = schwarz 1=blau 2=grün 3=türkis 4=rot 5=violett 6=gelb 7=weiß

Jeder weiße Balken besitzt außerdem immer eine Markierung an der X-Achse (die Achse ist unterbrochen, und in der Mitte der Unterbrechung befindet sich ein Punkt). Die weißen Balken inklusive der Markierungen dienen dazu, die Zuordnung von Grafikelementen zur Beschriftung der X-Achse zu kennzeichnen, die sich unterhalb des Grafikbereichs (der Ort entspricht Zeile 18, der letzten Zeile des Textbereiches) befindet.

Die Position der Y-Achse ist links von der Grafik festgelegt, die Achse ist alle 5 „Klötzchen“ markiert, besitzt also 19 feststehende Markierungen ( $90/5 + 1 = 19$ ). Jede Markierung der Y-Achse kann mit bis zu vier Zeichen Text beschriftet werden. Die Beschriftung ist ein String, der von der obersten Markierung beginnend, alle 4 Zeichen zur nächsten Markierung wechselt. Der gesamte String ist also  $19 \times 4 = 76$  Zeichen lang. Sollen nicht alle Markierungen beschriftet werden, so werden an dieser Stelle Leerzeichen eingebaut.

Die vertikale Position der X-Achse, d.h. der Nullpunkt der Y-Achse, ist beliebig festlegbar, sollte aber in der Regel an einer der 19 Markierungen der Y-Achse lokalisiert sein.

Es gibt 7 Grafikbefehle. Sie können angewendet werden, sofern der Grafik-Modus aktiviert ist. Der Befehl ESC[=h wechselt vom Text-Modus in den Grafikmodus. Da dieser Wechsel (wie auch beim Zurückwechseln in den Text-Modus) mit einer Löschung des gesamten Bildschirms verbunden ist, muß nach der Übermittlung dieses Befehls kurz gewartet werden (mindestens 20ms, besser 40 ms).

Die empfohlene Befehlssequenz für den Wechsel in den Grafikmodus lautet:

ESC[>5h        (Cursor unsichtbar schalten)  
ESC[?16h        (Textbildschirm dunkel schalten, erhöht die Löschgeschwindigkeit)  
ESC[2J           (Textbildschirm löschen)  
ESC[=h           (in den Grafikmodus wechseln)  
mindestens 20 mal das Zeichen „~“ (ASCII \$7E = dezimal 126 ) senden (ca. 20ms Pause)

Der Befehl ESC[2J (Textbildschirm löschen) ist eigentlich überflüssig, da bereits der Befehl ESC[=h die Löschaktion beinhaltet. Bei einem sehr „vollen“ Textbildschirm erfolgt jedoch der Wechsel in den Grafikmodus „weicher“. Alle Befehle können hintereinander weg ohne Pause übermittelt werden, nur am Schluss ist die Pause notwendig, während der Bildschirmbereich gelöscht wird. Da der explizite und der implizite Löschbefehl direkt hintereinander übermittelt werden, wird die Löschaktion nur ein Mal durchgeführt und dauert allenfalls unwesentlich länger als bei nur einem Löschbefehl.

Nun befinden wir uns im Grafikmodus. Der Grafikbildschirm (liegt im selben Speicherbereich wie der Textbildschirm) wurde gelöscht und zeigt nur die X-Achse an und nichts sonst. Im Grafikmodus funktionieren nur die sieben Grafikbefehle sowie ESC[=I (Rückkehr in den Text-Modus) sowie ESC[z (Warmstart-RESET). Die Grafikbefehle können nur über die serielle Schnittstelle übermittelt werden, die anderen beiden Befehle sowie CTRL-ALT-DEL (Warmstart) funktionieren auch über die Tastatur, sofern diese nicht für Kommandos gesperrt (ESC[>82h) oder komplett gesperrt (ESC[2h) ist, oder sich das Terminal im Remote-Modus befindet (dann kann der Host alle Tastatureingaben filtern, was ohnehin zu empfehlen ist).

Auch im Grafikmodus gilt Vollduplex-Mode, d.h. während Grafikdaten über die serielle Schnittstelle empfangen werden und die Grafik aktualisiert wird, werden alle Tastatureingaben über die serielle Schnittstelle herausgesendet (sofern nicht die Tastatur gesperrt ist).

Nun also kann die Grafik aufgebaut werden. Die sieben Grafikbefehle bestehen einfach aus den Ziffern 1..7 (ASCII \$31-\$37) und bedeuten:

### **„1“ – Grafik im 36-Balken-Modus aufbauen oder „2“ – Grafik im 72-Balken-Modus aufbauen**

Im 36-Balken-Modus belegen 36 statt 72 Balken die gesamte Horizontalausdehnung der Grafik. Dies ist (allein) wichtig, wenn bewegte „Echtzeitgrafik“ ausgegeben werden soll, da dann für die gesamte Grafik nur etwa halb so viele Daten benötigt werden. Im 36-Balken-Modus kann die Grafik mit über 20 Frames pro Sekunde aktualisiert werden (Einzelheiten siehe weiter unten).

Die beiden Befehle erwarten ein weiteres Byte als Parameter, nämlich die Lage des Nullpunktes der Y-Achse. Soll dieser genau in der (**M**)itte liegen, so entspricht dieser Wert dem ASCII-Code des Buchstabens „**M**“ (\$4D = dezimal 77), was wohl leicht zu merken ist. Der untersten Markierung der Y-Achse entspricht der Code für das Leerzeichen (\$20 = dezimal 32), der obersten der Code für den Kleinbuchstaben „**z**“ (\$7A = dezimal 122).

Nach dem Empfang dieser beide Bytes erscheint die X-Achse und das Terminal ist bereit für den Empfang des nächsten Grafikbefehls.

### **„3“ – Grafikdaten übermitteln**

Wird das Zeichen „3“ empfangen, während das Terminal auf Befehle wartet, dann werden gemäß den Werten der empfangenen Bytes die Balken chronologisch von links nach rechts aufgebaut. Der Wertebereich ist natürlich genau derselbe, wie bei der Definition des Nullpunktes der Y-Achse. Bei Werten kleiner als der Nullpunkt werden die Balken nach unten, ansonsten nach oben gezeichnet. Nachdem 36 oder 72 Balkenwerte empfangen wurden, schaltet das Terminal wieder auf Befehls-Empfangsbereitschaft. Die Datenübertragung kann auch schon vorher durch Übersendung des Zeichens „~“ (ASCII \$7E = dezimal 122) abgebrochen, und die Empfangsbereitschaft erreicht werden. Werden nunmehr ordnungsgemäß Daten im Bereich \$20-\$7A übertragen, so sieht man erst mal gar nichts, da anfänglich alle Farbwerte auf null (also schwarz) gesetzt sind. Dies kann man ändern und zwar mit dem nächsten Befehl

### **„4“ – Farbwerte definieren**

Nachdem der Befehl „4“ empfangen wurde, interpretiert das Terminal empfangene Bytes als neue Farbwerte für etwaig vorhandene Balken. Als Farbwert werden die drei niedrigstwertigen Bits verwendet. Praktischerweise passen diese Bits bei den ASCII-Codes für die Ziffern „0“-„7“ die Werte 0-7 (vermutlich kein Zufall). Bei den übertragenen Bytes spielen also die höchstwertigen 5 Bits keine Rolle. Auch hier kann, durch Übersendung des Zeichens „~“, die Übertragung abgebrochen, und in die Befehls-Empfangsbereitschaft zurückgeschaltet werden.

Zur Erinnerung hier noch einmal die Farbzuordnung:

0 = schwarz 1=blau 2=grün 3=türkis 4=rot 5=violett 6=gelb 7=weiß

Die Farbe weiß hat eine besondere Bedeutung, da am Ort jedes weißen Balkens auch die X-Achse markiert wird. Die Positionen der weißen Balken sollten sinnvollerweise mit der Beschriftung der X-Achse (siehe unten) korrespondieren.

### **„5“ – Kopfzeile übermitteln**

Im Anschluß an den Befehl „5“ kann ein aus maximal 50 Zeichen bestehender Textstring übermittelt werden. Kürzere Strings können wieder vorzeitig mit „~“ terminiert werden.

### **„6“ – Y-Achse beschriften**

Dem Befehl „6“ können bis zu 76 Zeichen folgen, welche die Beschriftung der Y-Achse ergeben. Die Beschriftung erfolgt von oben nach unten, wobei an jeder der 19 Markierungen vier Zeichen platziert werden. Soll beispielsweise nur jede zweite Markierung beschriftet werden, dann sind entsprechend Leerzeichen (\$20) in die Zeichenkette einzufügen.

### **„7“ – X-Achse beschriften**

Die dem Befehl „7“ nachfolgenden bis zu 50 Zeichen beschriften die X-Achse und befinden sich in der untersten Zeile des Bildschirmbereiches. Die Beschriftung der X-Achse und die in der Grafik enthaltenen weißen Balken (bzw. die Markierungen der X-Achse) sollten zueinander passend gewählt werden, damit die Grafik vom Betrachter leichter erfaßt werden kann.

### **Echtzeitgrafik**

Da die Grafik immer außerhalb des sichtbaren Bildschirmbereichs aktualisiert wird, können dynamische Veränderungen flimmerfrei dargestellt werden. Bei 9600 Baud 8N1, d.h. max. 960 Zeichen pro Sekunde und 36 Balken ergibt sich dank der effizienten Grafikbefehle eine rechnerische Framerate von  $960/37 = 25,95$  Frames pro Sekunde, da diese Daten dem Terminal ohne irgendwelche Wartezyklen übermittelt werden können. Dies ermöglicht beispielsweise die dynamische Darstellung von Daten eines Spektrum-Analyzers. Während dieser pausenlosen Datenübermittlung können alle Tastatureingaben parallel herausgesendet werden (Voll duplex).

### **Weniger Balken darstellen**

Wenige breite Balken können durch Gruppen gleichwertiger Balken in derselben Farbe dargestellt werden. Im 72-Balken-Modus sind 2,3,4,6,8,9,12,18,24 oder 36 Balken äquidistant bei voller Ausnutzung der Horizontalausdehnung darstellbar.

## **Sonstiges**

- Beim Einsatz des Programms „Superterm“ (ein sehr empfehlenswertes Freeware-Programm, herunterladbar von [www.mpfreezone.com](http://www.mpfreezone.com), das sich sehr gut auch zum Testen des TV-Terminal eignet), bewirken Betätigungen der ENTER-Taste beim PC lediglich einen Rücksprung des Cursors beim TV-Terminal zum Anfang derselben Zeile, da es in der Startkonfiguration für den Zeilenwechsel ein Linefeed benötigt, welches Superterm (zumindest in Version 2.14) nicht sendet. Mit Übersendung von ESC[>9I ESC[>8h an das TV-Terminal wird das Kommunikationsproblem behoben.