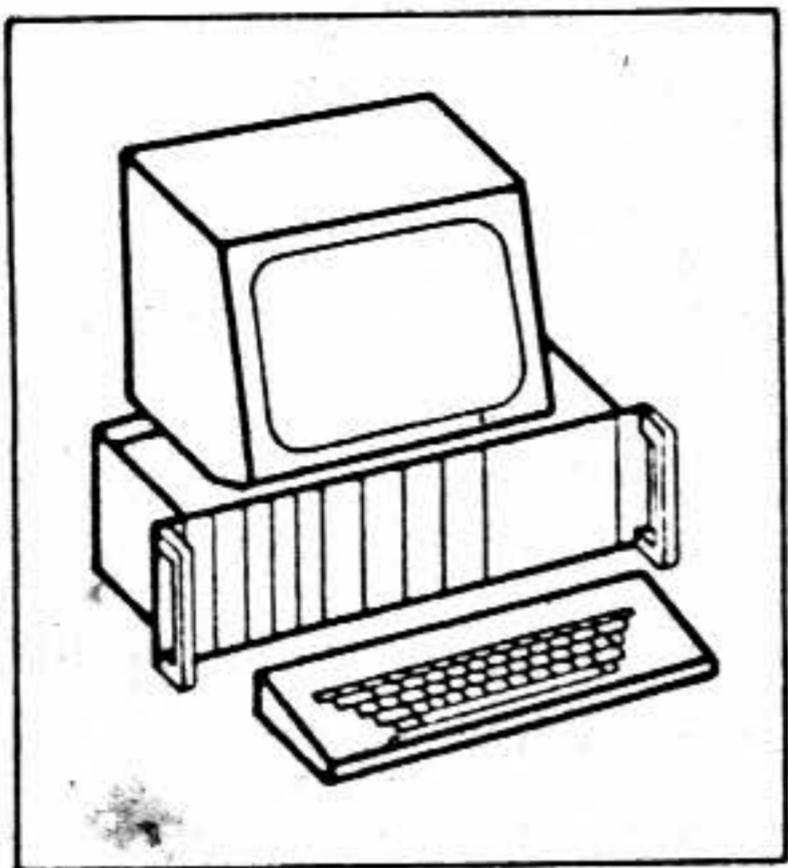


# FACHPRAKTISCHE ÜBUNG

# MIKROCOMPUTER-TECHNIK



## Video-Interface

BFZ/MFA 8.4.

Bildformat:  
24 Zeilen / 80 Zeichen



## Funktionsbeschreibung

### 1 Allgemeines

Das hier beschriebene Video-Interface BFZ/MFA 8.4 ersetzt das Video-Interface BFZ/MFA 8.2. Es bietet alle Funktionen des Interfaces BFZ/MFA 8.2 und kann darüber hinaus Kleinbuchstaben und Umlaute darstellen. Auf dem Video-Interface ist ein Summer integriert. Er übernimmt die Funktion des in der Tastatur eingebauten Summers. In der zusätzlichen Betriebsart "TVI 950" sind Steuersequenzen implementiert, die z.B. die Positionierung des Cursors und die Wahl verschiedener Darstellungsarten ermöglichen.

Das Video-Interface kann unter dem Betriebsprogramm MAT 85 und unter dem Betriebssystem CP/M verwendet werden. An dem Betriebsprogramm/Betriebssystem des MFA-Mikrocomputers sind keine Änderungen erforderlich. Eine Hardwareänderung (Drahtbrücken auf der verwendeten Serien-Schnittstelle) ist lediglich beim Einsatz unter dem Betriebssystem CP/M erforderlich. Grundvoraussetzung für ein ordnungsgemäßes Funktionieren (gleichgültig ob unter MAT 85 oder unter CP/M) ist allerdings die richtige Einstellung der DIL-Schalter auf dem Video-Interface BFZ/MFA 8.4. Diese Einstellung hängt vom verwendeten Betriebsprogramm ab und ist im Kapitel "Einsatz unter dem Betriebsprogramm MAT 85" bzw. im Kapitel "Einsatz unter dem Betriebssystem CP/M" beschrieben.

### 2 Funktionsbeschreibung

#### 2.1 Einleitung

Für den Austausch von Informationen zwischen Computer und Bediener verwendet man *Datensichtstationen*. In Computersystemen kommen Datensichtstationen meist dann zur Anwendung, wenn Textinformationen angezeigt oder eingegeben werden müssen.

Eine Datensichtstation besteht aus den Funktionseinheiten *Tastatur*, *Video-Interface* und *Monitor*. Der Monitor entspricht einem Fernsehgerät ohne HF-Empfangsteil, da das im Video-Interface erzeugte Bildsignal (auch BAS-Signal genannt) unmittelbar dem Monitor zugeführt wird. Bild 1 zeigt die Zusammenschaltung einer Datensichtstation mit einem Mikrocomputer.

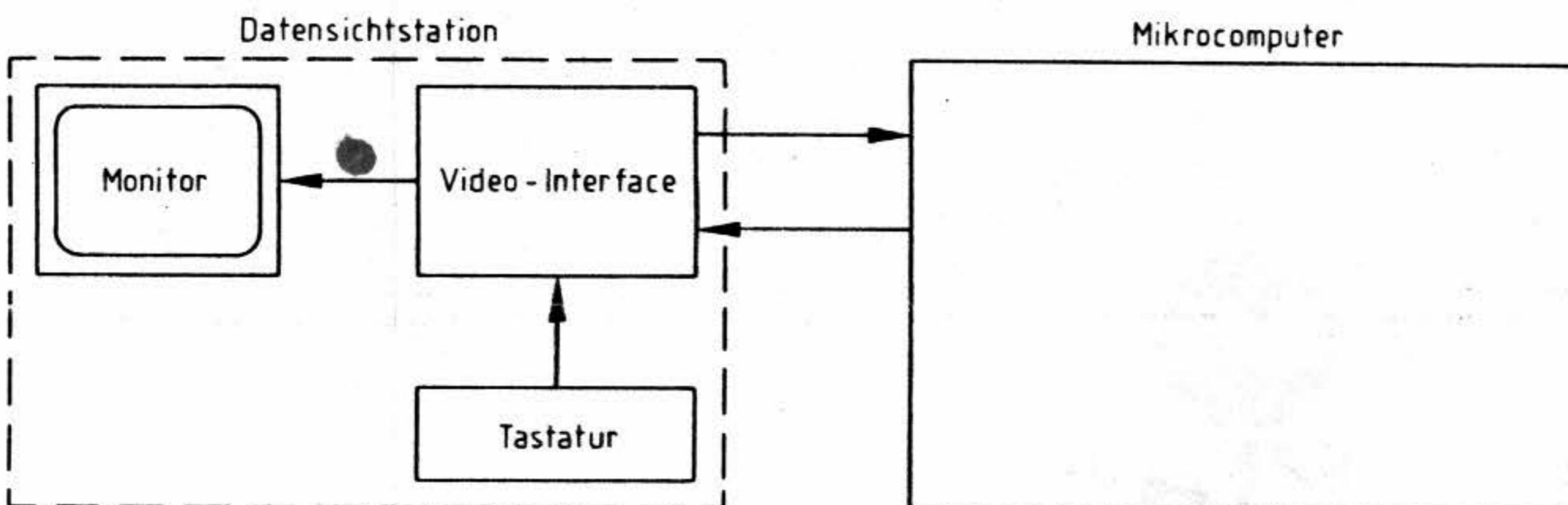


Bild 1: Datensichtstation und Mikrocomputer

## Funktionsbeschreibung

Jede Betätigung einer Taste der Tastatur bewirkt, daß ein Zeichen von der Datensichtstation an den Mikrocomputer gesendet wird. Andererseits werden Zeichen, die vom Mikrocomputer an die Datensichtstation gesendet werden, auf dem Bildschirm dargestellt. Für diesen Betrieb ist im Mikrocomputer ein Programm erforderlich, das Teil des Betriebssystems bzw. der Anwender-Software ist.

### 2.2 Darstellung von Zeichen auf dem Bildschirm

Für die Darstellung von Zeichen (Buchstaben, Ziffern, Sonderzeichen) auf dem Bildschirm wird ein Verfahren angewendet, bei dem jedes Zeichen durch Punkte eines *Matrixrasters* dargestellt wird. Häufig werden hierzu Matrixgrößen von 5x9 oder 7x10 Punkten verwendet. Bild 2 zeigt ein 5x9-Raster für die Buchstaben H und y.

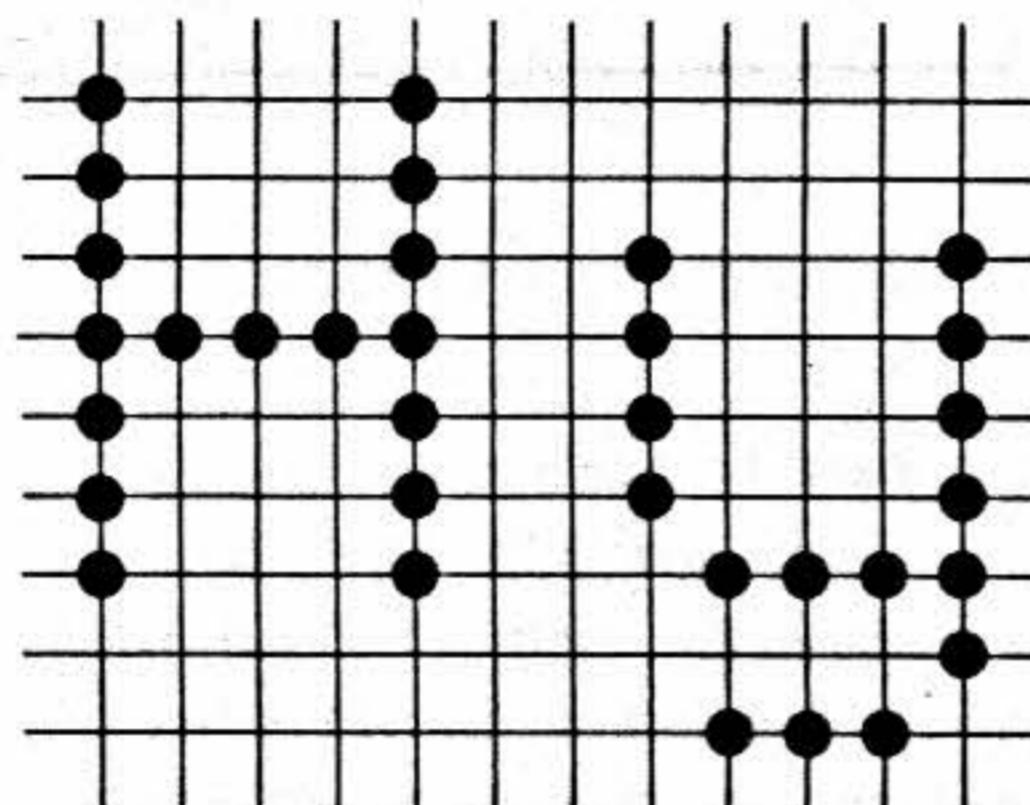


Bild 2: Zeichendarstellung im 5x9-Raster

Die einzelnen Punkte - auch *Pixel* genannt (vom engl. Picture Element, Bild-Element) - werden im Monitor durch einen Elektronenstrahl erzeugt, der entsprechend den darzustellenden Zeichenpunkten in seiner Intensität gesteuert wird (Hell-/Dunkelsteuerung). Dieser Strahl wandert kontinuierlich Punktreihe für Punktreihe über die Leuchtschicht der Bildröhre.

Zur Strahlablenkung enthält der Monitor zwei Sägezahngeneratoren. Die horizontale Ablenkung übernimmt ein Sägezahngenerator mit der Periodendauer  $T_1 = 64 \text{ us}$ . Die vertikale Ablenkung erfolgt durch einen zweiten Sägezahngenerator mit der Periodendauer  $T_2 = 20 \text{ ms}$ . Aus dem Verhältnis der beiden Periodenzeiten  $T_2/T_1$  ergibt sich die Anzahl von 312,5 Punktzeilen je Bild. Innerhalb einer Sekunde wird das gesamte Schriftfeld 50 mal auf dem Bildschirm dargestellt. Die Sägezahngeneratoren im Monitor werden durch Horizontal- und Vertikal-Impulse synchronisiert. Bild 3 zeigt Lage und Einteilung des Schriftfeldes auf dem Bildschirm.

## Funktionsbeschreibung

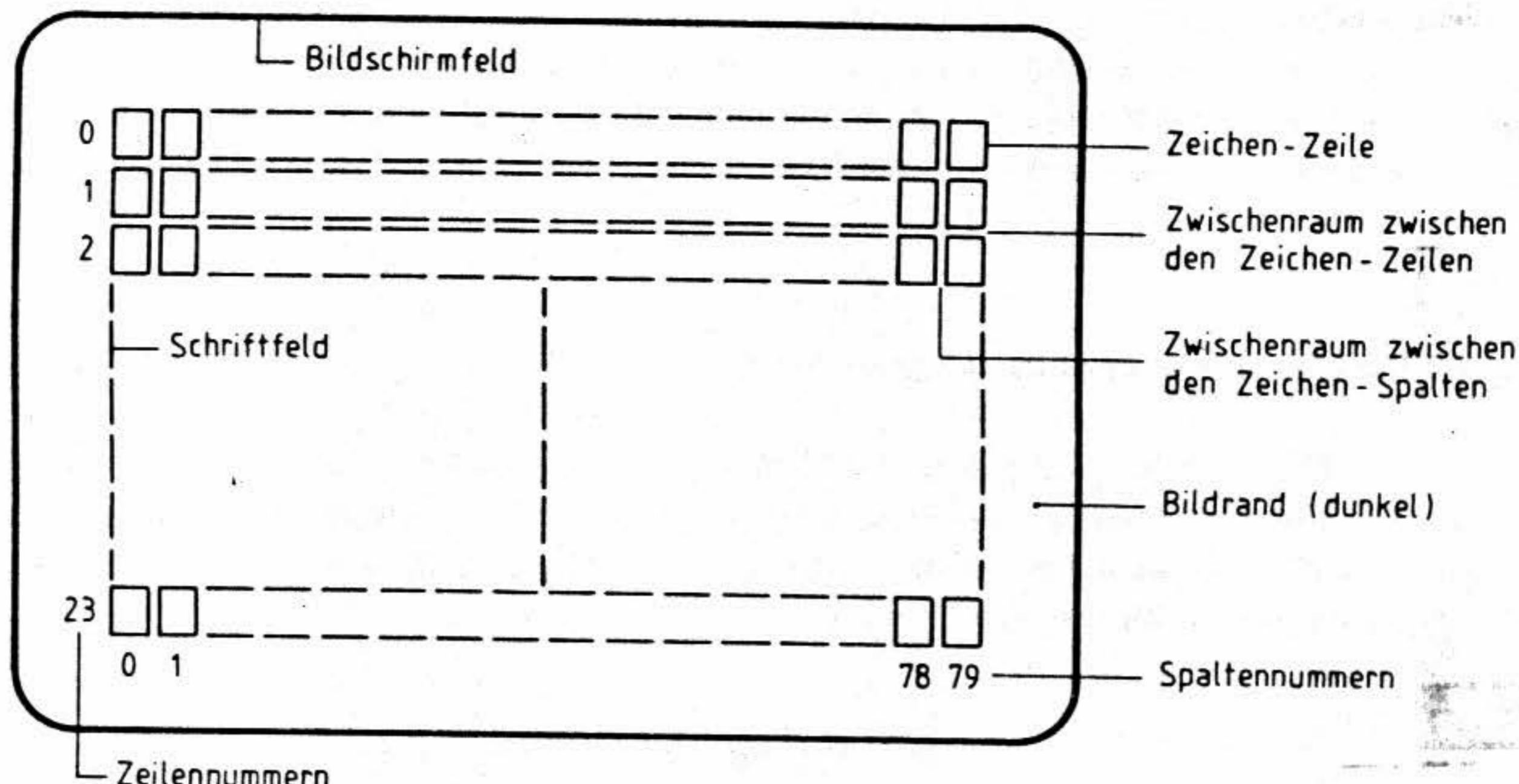


Bild 3: Schriftfeld auf dem Bildschirm

Das hier beschriebene Video-Interface kann wahlweise 22, 24, 26 oder 28 Zeichen-Zeilen und 72, 80, 88 oder 96 Zeichen-Spalten auf dem Bildschirm darstellen. Im folgenden wird jeweils von einer Einstellung auf 24 Zeichen-Zeilen und 80 Zeichen-Spalten ausgegangen. Diese Einstellung wurde ebenfalls für das Bild 3 angenommen. Zwischenräume zwischen den Zeichen sorgen für eine gute Lesbarkeit. Zur Vermeidung von Randunschärfen ist das Schriftfeld kleiner als das Bildschirmfeld. Dazu wird der Elektronenstrahl im Bereich der Bildschirmränder dunkelgesteuert.

Damit die einzelnen Zeichen – und damit das gesamte Schriftfeld – unverzerrt und ruhig auf dem Bildschirm erscheinen, muß das Video-Interface folgende Informationen an den Monitor liefern:

- Impulse, die dem Monitor anzeigen, daß ein neues Bild beginnen soll. Durch diese Impulse wird der Elektronenstrahl dunkelgesteuert und zur linken oberen Bildecke abgelenkt (Vertikal-Synchronimpulse).
- Impulse, die dem Monitor anzeigen, daß eine neue Punktzeile beginnen soll. Hierdurch wird der Elektronenstrahl dunkelgesteuert und zum nächsten Zeilenbeginn abgelenkt (Horizontal-Synchronimpulse).
- Die Information über die hell oder dunkel darzustellenden Bildpunkte einer Zeile (Bild-Signal).

Diese Informationen müssen entsprechend der CCIR-Fernsehnorm in einem Signal vermischt dem Monitor zugeführt werden. Dieses Signal nennt man kurz BAS-Signal, entsprechend der in ihm enthaltenen Informationen für Bild, Austastung (Dunkelsteuerung) und Synchronisierung.

### Funktionsbeschreibung

Bild 4 zeigt den Verlauf des Signals, das zur Darstellung der Punkte einer Punktreihe benötigt wird. 11 solcher "Signalpakete" werden zur Darstellung einer Zeichen-Zeile benötigt (9 zur Darstellung der jeweils 9 Pixel hohen Zeichen, 2 als Abstand zur nächsten Zeichen-Zeile). 264 "Signalpakete" werden zur Darstellung des gesamten Schriftfeldes vom Video-Interface erzeugt (24 Zeichen-Zeilen mit je 11 Punktspielen).

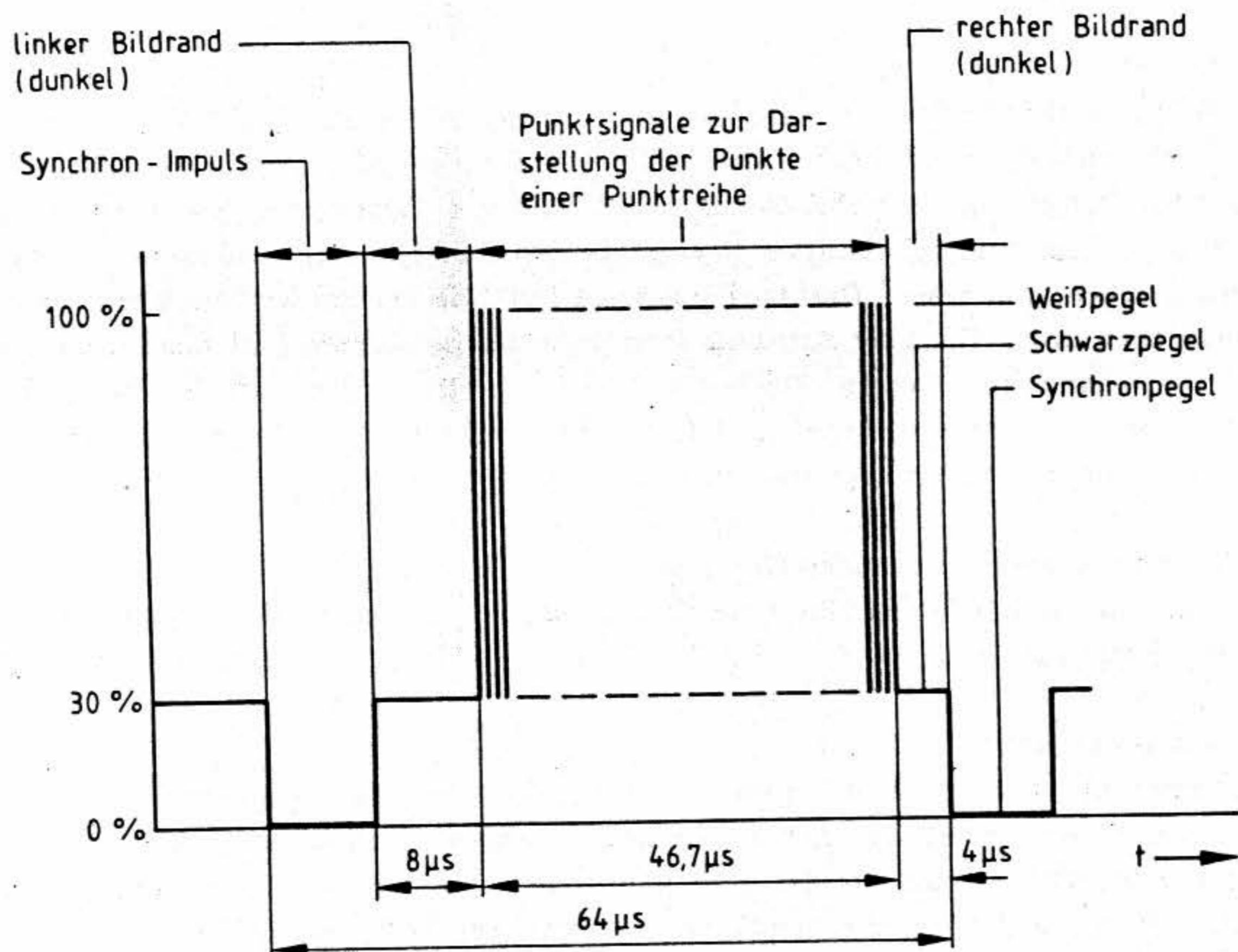


Bild 4: Das BAS-Signal

In einer Zeichen-Zeile werden 80 Zeichen (jeweils 5 Punkte breit) dargestellt. Zusammen mit den zwei Punkten zwischen den Zeichen ergeben sich so 560 Punkte je Punkt-Zeile. Zur Darstellung dieser Punkte innerhalb von 46,7  $\mu s$  benötigt man einen "Punktgenerator" mit einer Frequenz von 12 MHz.

## Funktionsbeschreibung

### 2.3 Die Erzeugung der Signale für die Zeichendarstellung

Das Blockschaltbild (Bild 5) zeigt die Haupt-Funktionseinheiten der Schaltung.

#### - Der CRT-Controller

Er steuert die Zeichenausgabe und erzeugt die hierfür notwendigen Signale. (CRT = cathode ray tube, Bildröhre. Controller = Steuerbaustein).

#### - Der Bildwiederhol-Speicher

Der Bildwiederhol-Speicher besteht aus dem *Zeichencode-Speicher* und dem *Attribut-Speicher*. Im Zeichencode-Speicher werden die Codes der auf dem Bildschirm darzustellenden Zeichen gespeichert. Durch die Speicher-Adresse innerhalb des Zeichencode-Speichers ist die Position des Zeichens auf dem Bildschirm festgelegt. Im Attribut-Speicher werden die *Attribute* der einzelnen Zeichen gespeichert. Die Attribute legen fest, wie ein Zeichen darzustellen ist (blinkend, unterstrichen usw.). Die über Attribute festgelegte *Darstellungsart* darf nicht mit der durch den Zeichengenerator festgelegten *Zeichenform* verwechselt werden. Der Zeichencode- und der Attribut-Speicher haben gemeinsame Adress- aber getrennte Daten-Leitungen. Am Zeichencode- und am Attribut-Speicher liegt also immer die gleiche Adresse an.

#### - Der Zeichengenerator (Character-Generator)

Der Zeichengenerator ist ein Nur-Lese-Speicher (ROM). Durch ihn wird die Form der einzelnen Zeichen festgelegt.

#### - Der Mikrocontroller

Der Mikrocontroller ist ein in einem IC zusammengefaßter Mikrocomputer (auch Single-Chip-Mikrocomputer genannt). Er wickelt den Datenaustausch zwischen Interface und MFA-Mikrocomputer ab. Der Mikrocontroller erkennt die vom Mikrocomputer zum Interface gesendeten Steuer-Zeichen und führt die gewünschten Funktionen (z.B. Bildschirm löschen) aus. Werden darstellbare Zeichen zum Interface übertragen, so "weiß" der Mikrocontroller, an welcher Stelle des Schriftfeldes die empfangenen Zeichen dargestellt werden sollen und schreibt sie in die entsprechende Speicher-Zeile des Bildwiederhol-Speichers ein.

#### 2.3.1 Einschreiben der Zeichen in den Bildwiederhol-Speicher

Die Zeichen, die auf dem Bildschirm dargestellt werden sollen, sind im ASCII-Code verschlüsselt (ASCII = American Standard Code for Information Interchange, amerikanischer Standard-Code für den Informationsaustausch). Eine ASCII-Codetabelle befindet sich im Anhang.

Das erste Zeichen, das der Mikrocomputer nach dem Einschalten zum Interface überträgt, wird in der linken oberen Ecke des Schriftfeldes dargestellt. Soll dort z.B. der Buchstabe "A" dargestellt werden, so sendet der Mikrocomputer den Code  $41_H$  an das Video-Interface. Der Microcontroller auf dem Interface erkennt aus dem Code, daß es sich um ein darstellbares Zeichen handelt. Er "weiß" außerdem, an welcher Position des Schriftfeldes das Zeichen dargestellt werden soll und

## Funktionsbeschreibung

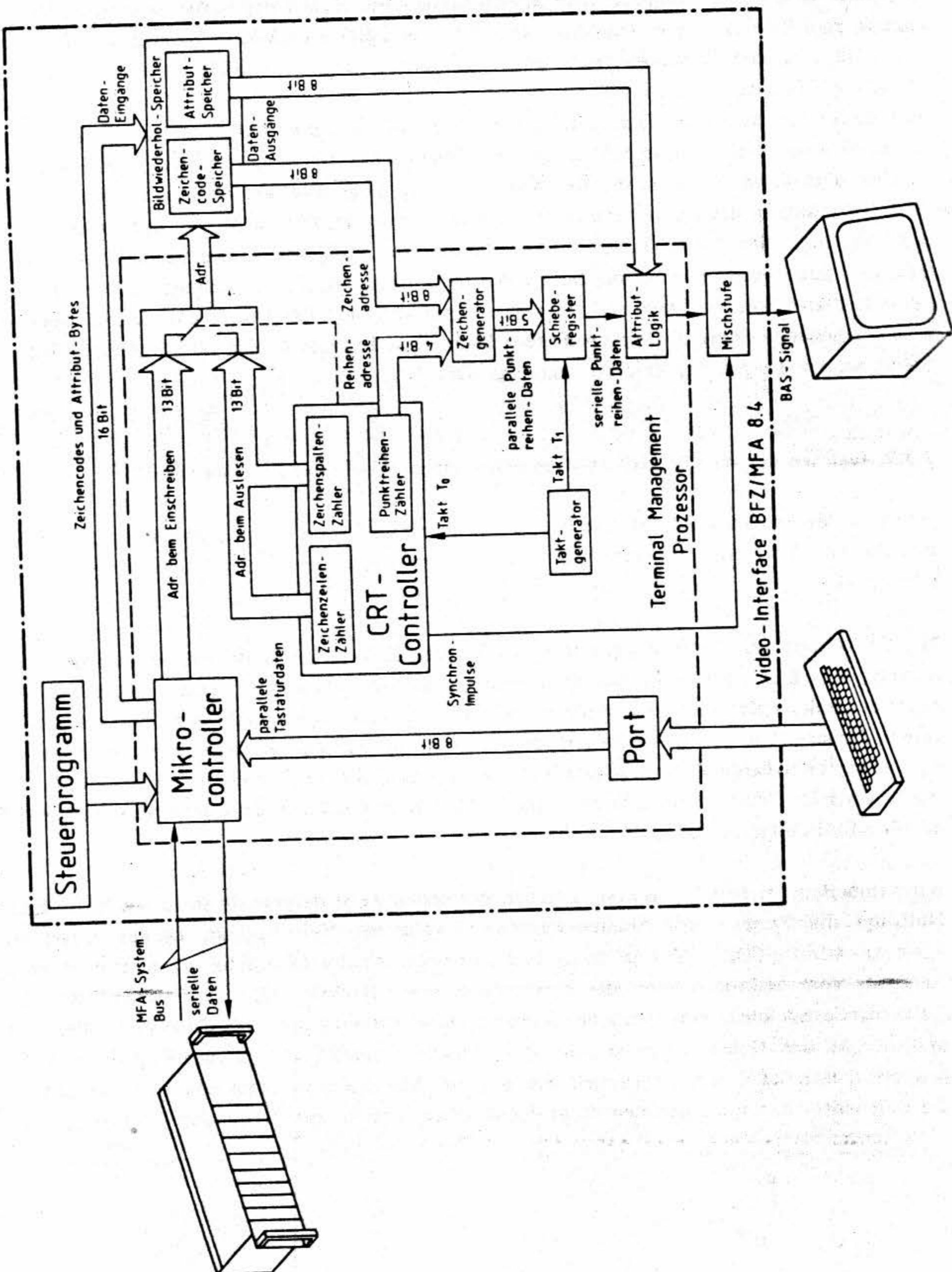


Bild 5: Blockschaltbild des Video-Interfaces

## Funktionsbeschreibung

trägt den Code in die entsprechende Speicherzeile (in diesem Falle in die erste Speicher-Zeile) des Zeichencode-Speichers ein. Überträgt der Mikrocomputer als nächstes Zeichen den Buchstaben "B" (Code  $42_H$ ) an das Video-Interface, so wird dieser Code in die nächste Speicher-Zeile des Zeichencode-Speichers geschrieben.

Hat der Mikrocomputer nach dem Einschalten zwei Zeichen (z.B. "A" und "B") zum Interface übertragen, so werden diese beiden Zeichen nebeneinander in der ersten Zeile des Schriftfeldes angezeigt. Das dritte Zeichen, das der Mikrocomputer zum Interface überträgt, würde rechts neben dem "B" - also in der dritten Spalte der ersten Zeile - angezeigt. Will man nun erreichen, daß dieses Zeichen in der dritten Spalte der zweiten Zeile angezeigt wird, so muß man zuvor das Steuer-Zeichen "Line Feed" (Zeilenvorschub, Code  $0A_H$ ) zum Interface übertragen. Der Mikrocontroller erkennt dieses Steuerzeichen und trägt das darauf folgende Zeichen in die richtige Speicher-Zeile des Zeichencode-Speichers ein. Auf dem Bildschirm kennzeichnet ein *Cursor* (Schreibmarke) die Stelle, an der das nächste Zeichen angezeigt wird.

### 2.3.2 Auslesen der im Bildwiederhol-Speicher eingeschriebenen Zeichen (Bilddarstellung)

Während der Elektronenstrahl eine Punktreihe überstreicht, muß er entsprechend den darzustellenden Zeichen hell- oder dunkelgesteuert werden. Der dazu notwendige Ablauf ist im folgenden beschrieben:

Der CRT-Controller besitzt intern mehrere Zähler. Einer davon zählt, welche Zeichen-Spalte momentan vom Elektronenstrahl auf dem Bildschirm geschrieben wird (Zeichenspalten-Zähler). Ein weiterer zählt, welche Zeichen-Zeile momentan vom Elektronenstrahl geschrieben wird (Zeichenzeilen-Zähler). Um einen Zeichen-Code aus dem Zeichencode-Speicher auszulesen, schaltet der CRT-Controller beide Zählerstände auf die Adressleitungen des Speichers (siehe Bild 5). Auf diese Weise wird im Zeichencode-Speicher das Zeichen adressiert, das momentan vom Elektronenstrahl auf den Bildschirm angezeigt wird.

Wird zum Beispiel gerade das erste Zeichen der ersten Zeile dargestellt, so stehen beide Zähler auf Null, da die Zeilen- und Spalten-Zählung jeweils bei Null beginnt. Daraus ergibt sich die Speicher-Adresse  $0000_H$ . Soll an dieser Bildschirmposition der Buchstabe "A" dargestellt werden, so ist zuvor vom Mikrocontroller der entsprechende ASCII-Code ( $41_H$ ) in diese Speicher-Zeile des Zeichencode-Speichers eingeschrieben worden. Der Code wird nun vom CRT-Controller ausgelesen und liegt an den Daten-Ausgängen des Zeichencode-Speichers an. Durch diesen Code ist lediglich festgelegt, welches Zeichen dargestellt werden soll. Wie dieses Zeichen dargestellt werden soll, also die Zeichenform, wird durch den nachfolgend beschriebenen Zeichengenerator festgelegt.

## Funktionsbeschreibung

### 2.3.3 Auslesen der Zeichenform-Information aus dem Zeichengenerator

Die Information über die Form des darzustellenden Zeichens ist im Zeichengenerator festgelegt. Hierbei handelt es sich um einen Nur-Lese-Speicher (ROM). Bild 6 zeigt den Inhalt einiger Speicher-Zeilen dieses ROMs. Die dargestellten Speicher-Zeilen enthalten die Information über die Form der Buchstaben "G" und "g". Die aus dem Zeichencode-Speicher ausgelesenen Daten (ASCII-Code der darzustellenden Zeichen) gelangen auf die höherwertigen Adreßleitungen des Zeichengenerators. Über diese Adreßleitungen wird die Zeichenform-Information für das darzustellende Zeichen ausgewählt. Man spricht daher auch von der *Zeichen-Adresse*.

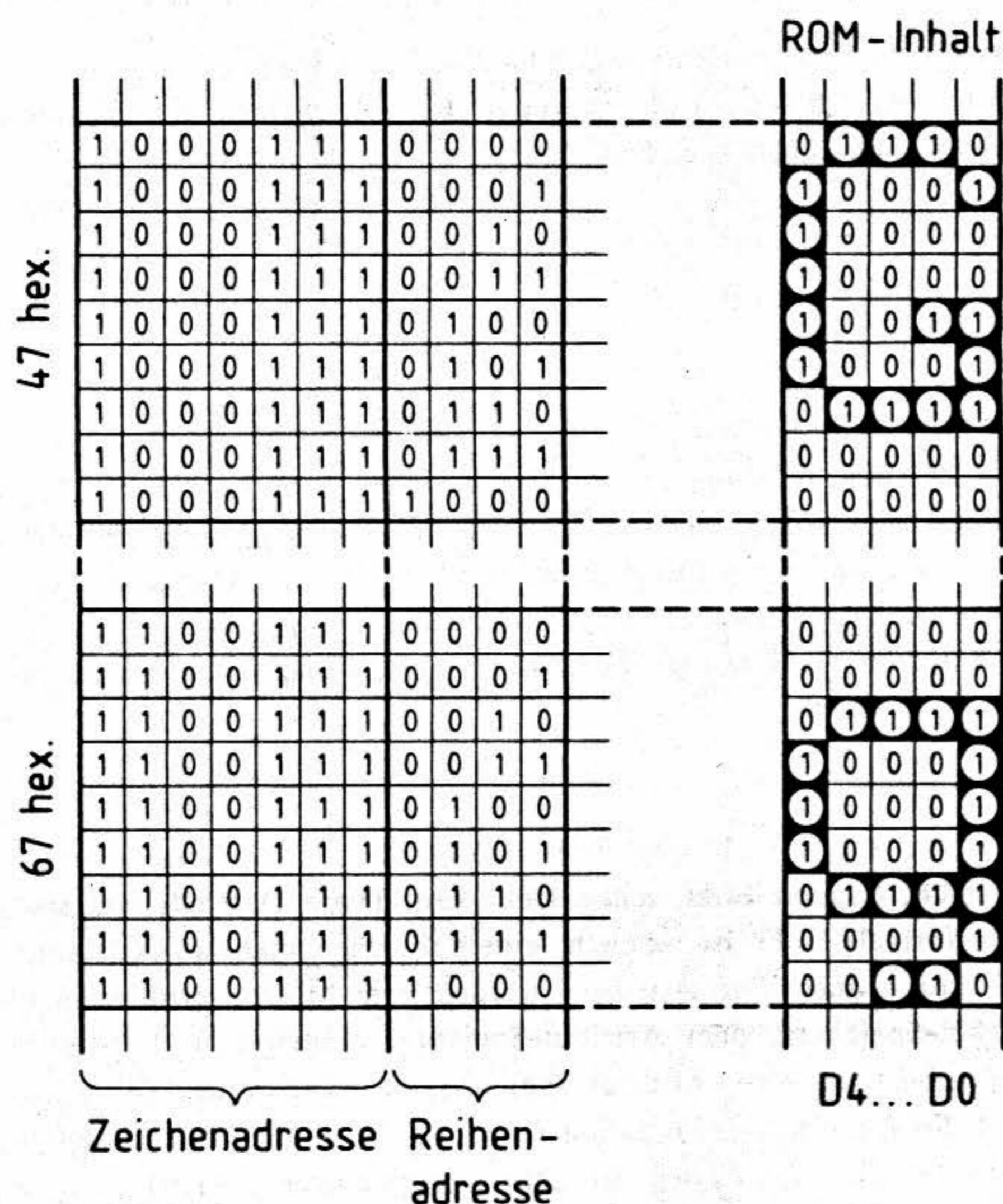


Bild 6: Programmierung des Zeichengenerators (Auszug)

Die einzelnen Zeichen sind aus 9 Punktreihen aufgebaut. Während der Elektronenstrahl von links nach rechts über den Bildschirm abgelenkt wird, zeichnet er jeweils eine dieser Punktreihen. Ein weiterer Zähler innerhalb des CRT-Controllers enthält hierzu die Information, in welcher Punktreihe des Zeichens der Elektronenstrahl sich momentan befindet. Der Stand dieses Punktreihen-Zählers gelangt auf die niederwertigen Adressleitungen des Zeichengenerators (Reihen-Adresse). Über die Reihenadresse wird die Zeichenform-Information einer Punktreihe (Punkt-Adresse).

## Funktionsbeschreibung

reihen-Daten) des darzustellenden Zeichens ausgewählt. Die Daten einer Punktreihe werden aus dem Zeichengenerator ausgelesen und gelangen zum Schieberegister.

### 2.3.4 Erzeugung des BAS-Signals aus den Punktreihen-Daten

Zur Ansteuerung des Monitors muß aus den Punktreihen-Daten noch das BAS-Signal gewonnen werden. Hierzu setzt ein Schieberegister die Punktreihen-Daten in einen seriellen Datenstrom um. Während der Elektronenstrahl die Zeichen-Spalte überstreicht, werden nacheinander alle 5 Punkte der Punktreihe des darzustellenden Zeichens am Ausgang des Schieberegisters ausgegeben.

Das Schieberegister liefert aber nur den Bild-Anteil des BAS-Signals. Die Austast- und Synchron-Impulse werden vom CRT-Controller erzeugt. Die Mischstufe vereinigt alle Signal-Anteile zum BAS-Signal (siehe auch Bilder 4 und 5).

## 2.4 Steuerzeichen und Attribute

### - Steuerzeichen

Um das Video-Interface vielseitig nutzen zu können, überträgt der Mikrocomputer außer den darzustellenden Zeichen auch sogenannte Steuerzeichen zum Interface. Eines dieser Steuerzeichen ist "Carriage Return" mit dem Code  $0D_H$ . Nach der Übertragung dieses Steuerzeichens wird das nächste darstellbare Zeichen am linken Rand des Schriftfeldes angezeigt. Die Steuerzeichen werden nicht in den Bildwiederhol-Speicher eingetragen. Der Mikrocontroller des Video-Interfaces filtert diese Zeichen aus dem empfangenen Datenstrom heraus und führt die entsprechenden Funktionen aus.

### - Attribute

Das Video-Interface bietet vielfältige Darstellungsarten der Zeichen: invers, unterstrichen, blinkend, doppelt hoch, doppelt breit, halbe Helligkeit. Diese Darstellungsarten werden durch sogenannte *Attribute* festgelegt. Es ist möglich, einem Zeichen mehrere Attribute - z.B.: unterstrichen und blinkend - zuzuordnen. Die Attribute werden vom Mikrocontroller in einen besonderen Teil des Bildwiederhol-Speichers, dem Attribut-Speicher, eingetragen. In diesem Speicher ist jeder Zeichencode-Speicherzeile eine Attribut-Speicherzeile zugeordnet. In jedem Bit der Attribut-Speicherzeile ist die Ein-/Aus-Information für eines der möglichen Darstellungsarten festgehalten. Theoretisch können einem Zeichen so acht verschiedene Attribute zugeordnet werden - die Steuersoftware des Mikrocontrollers unterstützt jedoch nur sieben.

Die Attribute werden über Steuerzeichen-Folgen ein- bzw. ausgeschaltet. Die eingeschalteten Attribute gelten für alle Zeichen, die bis zum Ausschalten der entsprechenden Attribute übertragen werden.

## Funktionsbeschreibung

### Beispiel:

1. Steuerzeichen-Folge: NUR "UNTERSTREICHEN" EINSCHALTEN  
*Alle folgenden Zeichen werden unterstrichen dargestellt*
2. Steuerzeichen-Folge: ZUSÄTZLICH "BLINKEN" EINSCHALTEN  
*Alle folgenden Zeichen werden unterstrichen und blinkend dargestellt*
3. Steuerzeichen-Folge: "UNTERSTREICHEN" AUSCHALTEN (andere Attribute unverändert)  
*Alle folgenden Zeichen blinken, werden aber nicht unterstrichen dargestellt*
4. Steuerzeichen-Folge: "BLINKEN" AUSCHALTEN  
*Alle folgenden Zeichen werden nun normal (nicht blinkend) dargestellt*

Der Mikrocontroller erkennt die Steuerzeichen zum Ein- bzw. Ausschalten der Attribute und merkt sich die aktuellen Zustände (ein/aus) der Attribute. Immer wenn er einen Zeichen-Code in den Zeichencode-Speicher einträgt, schreibt er parallel dazu die Zustände der Attribute in den Attribut-Speicher.

Soll ein im Bildwiederhol-Speicher befindliches Zeichen auf dem Bildschirm dargestellt werden, so wird der Zeichen-Code aus dem Zeichencode-Speicher ausgelesen. Parallel dazu wird das zugehörige Attribut-Byte aus dem Attribut-Speicher ausgelesen. Das Attribut-Byte steuert die *Attribut-Logik*. Diese Logik befindet sich im Signalweg zwischen Schieberegister und Mischstufe (siehe auch Bild 5). Bei Zeichen mit dem Attribut "INVERS" sorgt diese Logik z.B. dafür, daß die einzelnen Punktreihen-Daten des Zeichens invertiert werden.

## 2.5 Die Gesamtschaltung des Video-Interfaces

Der Stromlaufplan im Anhang zeigt die Gesamtschaltung des Video-Interfaces.

Der Mikrocontroller, sein interner RAM-Speicher (64x8 Bit), der CRT-Controller, der Zeichen-generator, die Attributlogik, das Schieberegister und der Taktgenerator befinden sich zusammen auf einem Chip - dem Terminal-Management-Processor (TMP) NS 405 A12N. Zusätzlich zu den erwähnten Funktionsgruppen befinden sich noch ein paralleler 8-Bit-I/O-Port und ein Baudraten-generator im TMP.

Der Bildwiederhol-Speicher besteht aus zwei RAM-Bausteinen des Typs 6264 (D2 und D3, Speicherkapazität: je 8-K-Byte) für Zeichencode- und Attribut-Speicher. Als Programm-Speicher für den im TMP integrierten Mikrocontroller dient ein 8-K-Byte-EPROM des Typs 2764 (D11). Die Mischstufe wird aus dem Antivalenz-Gatter D16, dem Transistor V1, der Diode V3 und einigen Widerständen gebildet.

Da die Übertragung der Daten zwischen Mikrocomputer und Interface seriell mit V.24-Pegel erfolgt, besitzt das Interface noch zwei Pegelwandler zur Wandlung von TTL- in V.24-Pegel und umgekehrt (V2 und D14). Außerdem befinden sich drei Schalterreihen auf der Baugruppe. Über diese Schalter können das Übertragungsformat zwischen Interface und Mikrocomputer sowie die verschiedenen Betriebsarten des Interfaces gewählt werden.

## MAT 85-Mode

**3 Emulation des Video-Interfaces BFZ/MFA 8.2 und des Terminals TVI 950**

Auf dem Markt gibt es verschiedene Datensichtstationen. Diese arbeiten teilweise mit unterschiedlichen Steuerzeichen. Das hier beschriebene Video-Interface kann in zwei verschiedenen Modi arbeiten. In jedem dieser Modi wird eine andere Datensichtstation emuliert (nachgebildet). Im ersten Mode (dem MAT 85-Mode) wird das Video-Interface BFZ/MFA 8.2 mit seinen Steuerzeichen, im zweiten Mode (dem TVI 950-Mode) wird das Terminal TVI 950 der Firma TeleVideo emuliert. In beiden Fällen handelt es sich um eine Teil-Emulation. So werden z.B. im MAT 85-Mode immer mehr Zeichen-Zeilen und Zeichen-Spalten als im emulierten Video-Interface angezeigt (die Anzahl der Zeilen und Spalten ist einstellbar). Im TVI 950-Mode werden nur die wichtigsten Steuerzeichen dieses Terminals unterstützt.

Über den Schalter S1.1 der Interface-Baugruppe kann eingestellt werden, welcher Mode nach dem Einschalten aktiv sein soll. Der Mode kann während des Betriebs beliebig häufig über die Tastatur oder durch Steuerzeichen gewechselt werden.

In den folgenden Kapiteln wird zwischen darstellbaren Zeichen und Steuerzeichen unterschieden. Überträgt man den Code eines darstellbaren Zeichens vom Mikrocomputer zum Interface, wird das entsprechende Zeichen auf dem Bildschirm angezeigt. Hierbei ist die Art der Zeichendarstellung vom eingestellten *Zeichensatz* abhängig. Die Wahl des Zeichensatzes legt z.B. fest, ob deutsche oder französische Sonderzeichen angezeigt werden sollen. Überträgt man den Code eines Steuerzeichens zum Interface, wird die entsprechende Steuerfunktion ausgeführt.

In der Beschreibung der einzelnen Steuerfunktionen wird häufig der Begriff "Cursor" verwendet. Hierbei handelt es sich um eine Schreibmarke, die dem Anwender anzeigen, an welcher Stelle des Schriftfeldes das nächste zum Interface übertragene Zeichen dargestellt wird. Die Form dieser Schreibmarke (Strich oder Block) kann eingestellt werden.

**3.1 Der MAT 85-Emulations-Mode**

Dieser Mode ist für den Betrieb unter dem Betriebsprogramm MAT 85 des MFA-Mikrocomputers vorgesehen. In diesem Mode können alle Steuerzeichen des Video-Interfaces BFZ/MFA 8.2 benutzt werden. Bei vier dieser Steuerzeichen reagiert das hier beschriebene Interface (BFZ/MFA 8.4) geringfügig anders. Dies führt jedoch i. allg. zu keinerlei Problemen.

Zusätzlich zu den oben erwähnten Steuerzeichen, "kennt" das hier beschriebene Interface weitere Steuerzeichen-Folgen (Sequenzen). Mit diesen Sequenzen kann man z.B. von einem Mode in den anderen schalten oder einen Software-Reset (Kaltstart) auslösen.

Unter dem Betriebssystem CP/M sollte der TVI 950-Mode dem MAT 85-Mode vorgezogen werden, da dieser mehr Möglichkeiten (z.B. direkte Cursor-Positionierung) bietet.

---

MAT 85-Mode

### 3.1.1 Die 1-Byte-Steuerzeichen im MAT 85-Mode

Die Steuerfunktionen des Interfaces werden im MAT 85-Mode sowohl durch einzelne Steuerzeichen als auch durch Steuerzeichen-Folgen ausgelöst. In diesem Kapitel sind die einzelnen Steuerzeichen (Control-Codes) beschrieben. Eine Beschreibung der Steuerzeichen-Folgen (Sequenzen) finden Sie im nächsten Kapitel.

Um die entsprechende Steuerfunktion auszulösen, muß der Code des Steuerzeichens vom Mikrocomputer zum Video-Interface übertragen werden. Näheres dazu entnehmen Sie bitte dem Kapitel "Einsatz unter dem Betriebsprogramm MAT 85".

Für jedes Steuerzeichen ist im folgenden seine Bezeichnung, sein hexadezimaler Code, sein dezimaler Code und seine Funktion angegeben.

#### Summer

**BEL**  $07_H$   $7_D$   
Durch die Ausgabe des Steuerzeichens BEL ertönt der im Interface eingebaute Summer für etwa 0,5 Sekunden.

---

#### Cursor nach links (Backspace)

**BS**  $08_H$   $8_D$   
Der Cursor wird um eine Spalte nach links bewegt. Befindet er sich am linken Schriftfeld-Rand, so bewegt er sich an das Ende der vorstehenden Zeile. Wenn der Cursor sich in der linken oberen Ecke des Schriftfeldes befindet, hat dieses Steuerzeichen keine Wirkung.

Dieses Steuerzeichen hat bei dem hier beschriebenen Interface eine geringfügig andere Wirkung als beim Video-Interface BFZ/MFA 8.2: Befindet sich der Cursor beim Interface BFZ/MFA 8.2 links oben im Schriftfeld, so bewegt er sich durch das Steuerzeichen BS zur rechten unteren Ecke.

---

#### Cursor nach rechts

**HT**  $09_H$   $9_D$   
Der Cursor wird um eine Spalte nach rechts bewegt. Steht der Cursor am Ende einer Zeile, springt er an den Anfang der nächsten Zeile. Wenn der Cursor rechts unten steht, wird das Schriftfeld um eine Zeile nach oben geschoben (gescrollt). Der Inhalt der obersten Zeichen-Zeile geht verloren, unten wird eine Leerzeile eingefügt. Der Cursor springt an den Anfang dieser Leerzeile.

Dieses Steuerzeichen hat bei dem hier beschriebenen Interface eine geringfügig andere Wirkung als beim Video-Interface BFZ/MFA 8.2: Beim Interface 8.2 wird die oben herausgeschobene Zeichen-Zeile anstelle der Leerzeile unten eingefügt.

## MAT 85-Mode

**Cursor nach unten (Line Feed, Zeilenvorschub)**

✓ LF

0A<sub>H</sub>10<sub>D</sub>

Der Cursor wird - unter Beibehaltung der Spaltenposition - um eine Zeile nach unten bewegt. Steht der Cursor am unteren Schriftfeld-Rand, wird das Schriftfeld um eine Zeile nach oben geschoben. Dabei geht der Inhalt der obersten Zeichen-Zeile verloren. Unten wird eine Leerzeile eingefügt.

**Cursor nach oben**

✓ VT

0B<sub>H</sub>11<sub>D</sub>

Der Cursor wird - unter Beibehaltung der Spaltenposition - um eine Zeile nach oben bewegt. Steht der Cursor am oberen Schriftfeld-Rand, wird dieses Steuerzeichen ignoriert.

Dieses Steuerzeichen hat bei dem hier beschriebenen Interface eine geringfügig andere Wirkung als beim Video-Interface BFZ/MFA 8.2: Befindet sich der Cursor beim Interface 8.2 am oberen Schriftfeld-Rand, springt er - unter Beibehaltung der Spaltenposition - an den unteren Schriftfeldrand.

**Bildschirm löschen, Cursor in Home-Position**

✓ FF

0C<sub>H</sub>12<sub>D</sub>

Der Bildschirm wird mit Leerzeichen gefüllt (gelöscht). Der Cursor wird in die obere linke Ecke des Schriftfeldes bewegt (Home-Position).

**Cursor an den Zeilenanfang, Zeilenrest löschen**

✓ CR

0D<sub>H</sub>13<sub>D</sub>

Der Cursor wird an den Zeilenanfang bewegt. Die Zeichen ab der alten Cursor-Position (einschließlich) bis zum Zeilenende werden durch Leerzeichen überschrieben (gelöscht). Wenn der Cursor bereits am Zeilenanfang steht, hat dieses Steuerzeichen keine Wirkung.

**Zeile löschen**

✓ SUB

1A<sub>H</sub>26<sub>D</sub>

Die Zeile, in der sich der Cursor befindet, wird komplett mit Leerzeichen gefüllt (gelöscht). Die Cursor-Position bleibt erhalten.

## MAT 85-Mode

## Cursor nach unten

**ESC**  $1B_H$   $27_D$  ✓

Der Cursor wird - unter Beibehaltung der Spaltenposition - um eine Zeile nach unten bewegt. Steht der Cursor am unteren Schriftfeld-Rand, wird der Inhalt des Schriftfeldes um eine Zeile nach oben geschoben (gescrollt). Hierbei geht der Inhalt der obersten Zeile verloren. Unten wird eine Leerzeile eingefügt.

Dieses Steuerzeichen hat bei dem hier beschriebenen Interface eine geringfügig andere Wirkung als beim Video-Interface 'BFZ/MFA 8.2: Befindet sich der Cursor beim Interface 8.2 am unteren Schriftfeld-Rand, wird die oben herausgeschobene Zeile unten wieder eingefügt.

## Cursor in die Home-Position

**FS** *file separator*  $1C_H$   $28_D$  ✓

Der Cursor wird in die linke obere Ecke des Schriftfeldes (Home-Position) gestellt. Hierbei wird der Bildschirm nicht gelöscht.

## Cursor an den Zeilenanfang

**GS** *group Separator*  $1D_H$   $29_D$  ✓

Der Cursor wird an den Zeilenanfang bewegt.

## 3.1.2 Die Steuer-Sequenzen im MAT 85-Mode

Im MAT 85-Mode kennt das Interface zusätzlich zu den 1-Byte-Steuerzeichen weitere fünf Steuer-Sequenzen. Diese Sequenzen beginnen alle mit dem Zeichen DLE (Data Link Escape,  $10_H$ ) und bestehen aus jeweils drei Zeichen. Um eine Steuerfunktion auszulösen, müssen die Codes der entsprechenden drei Zeichen vom Mikrocomputer zum Interface übertragen werden. Die DLE-Sequenzen sind identisch mit den DLE-Sequenzen im TVI 950-Mode.

Für jede Sequenz ist im folgenden ihre Bezeichnung, ihre hexadezimale Code-Folge, ihre dezimale Code-Folge und ihre Funktion angegeben.

## TVI 950-Mode einschalten

**DLE DLE 1**  $10_H 10_H 31_H$   $16_D 16_D 49_D$

Mit dieser Sequenz kann in den TVI 950-Mode geschaltet werden. Nach der Umschaltung gelten die Steuerzeichen-Definitionen des TVI 950-Modes.

**MAT 85-Mode****MAT 85-Mode einschalten****DLE DLE 2** $10_{\text{H}} \ 10_{\text{H}} \ 32_{\text{H}}$  $16_{\text{D}} \ 16_{\text{D}} \ 50_{\text{D}}$ 

Mit dieser Sequenz kann in den MAT 85-Mode geschaltet werden. Befindet man sich bereits im MAT 85-Mode, so wird diese Sequenz ignoriert.

**Aktuellen Mode abfragen****DLE DLE ?** $10_{\text{H}} \ 10_{\text{H}} \ 3F_{\text{H}}$  $16_{\text{D}} \ 16_{\text{D}} \ 63_{\text{D}}$ 

Mit Hilfe dieser Sequenz kann der Anwender abfragen, welcher Mode (MAT 85 oder TVI 950) momentan aktiv ist. Nach dem Empfang dieser Sequenz "antwortet" das Interface mit drei Bytes:

Antwort-Bytes	aktiver Mode
$4D_{\text{H}} \ 31_{\text{H}} \ 0D_{\text{H}}$	TVI 950-Mode
$4D_{\text{H}} \ 32_{\text{H}} \ 0D_{\text{H}}$	MAT 85-Mode

Beachten Sie hierzu bitte auch das Kapitel "Einsatz unter dem Betriebsprogramm MAT 85".

**Software-Reset****DLE DLE @** $10_{\text{H}} \ 10_{\text{H}} \ 40_{\text{H}}$  $16_{\text{D}} \ 16_{\text{D}} \ 64_{\text{D}}$ 

Durch diese Sequenz kann ein sogenannter Software-Reset des Video-Interfaces ausgelöst werden. Das heißt, das Interface wird so eingestellt, wie es nach dem Anlegen der Betriebsspannung eingestellt sein würde.

**Abfrage der Software-Version****DLE DLE V** $10_{\text{H}} \ 10_{\text{H}} \ 56_{\text{H}}$  $16_{\text{D}} \ 16_{\text{D}} \ 86_{\text{D}}$ 

Mit Hilfe dieser Sequenz kann der Anwender die Versionsnummer der Software im EPROM der Interface-Baugruppe abfragen. Die Version ist durch zwei Ziffern gekennzeichnet (Versions-Nummer). Nach dem Empfang dieser Sequenz "antwortet" das Interface mit fünf Bytes:

Beispiel für eine Antwort bei Version 1.0:

$56_{\text{H}} \ 31_{\text{H}} \ 2F_{\text{H}} \ 30_{\text{H}} \ 0D_{\text{H}}$

Das erste Byte der Antwort ist immer  $56_{\text{H}}$  (ASCII "V").

Das zweite Byte entspricht dem ASCII-Code der ersten Ziffer der Versions-Nummer (hier "1").

Das dritte Byte ist (bei dieser Interface-Hardware) immer  $2F_{\text{H}}$ .

Das vierte Byte entspricht dem ASCII-Code der zweiten Ziffer der Versions-Nummer (hier "0").

Das fünfte Byte der Antwort ist immer  $0D_{\text{H}}$ .

Beachten Sie hierzu bitte auch das Kapitel "Einsatz unter dem Betriebsprogramm MAT 85".

## TVI 950-Mode

## 3.2 Der TVI 950-Emulations-Mode

Der TVI 950-Mode ist vorwiegend für den Einsatz unter dem Betriebssystem CP/M vorgesehen. Die Steuerzeichen dieses Modes unterscheiden sich teilweise von denen des MAT 85-Modes. Deshalb kann der TVI 950-Mode nur eingeschränkt unter dem Betriebsprogramm MAT 85 benutzt werden. Allerdings "kennt" das Interface im TVI 950-Mode mehr Steuerzeichen. So ist es in diesem Mode z.B. möglich, den Cursor durch Angabe einer Zeilen- und Spalten-Nummer an eine bestimmte Stelle des Schriftfeldes zu positionieren. Außerdem können die Zeichen auf vielfältige Art dargestellt werden: invers, unterstrichen, blinkend, doppelt hoch und doppelt breit.

Aufgrund seiner vielen Möglichkeiten werden Steuerfunktionen im TVI 950-Mode oft nicht durch einzelne Steuerzeichen, sondern durch Steuerzeichen-Folgen (Sequenzen) ausgelöst. Beginnen diese Sequenzen mit dem Zeichen ESC (Escape, Code  $1B_H$ ), so spricht man auch von einer *Escape-Sequenz*.

## 3.2.1 Die 1-Byte-Steuerzeichen im TVI 950-Mode

In diesem Kapitel sind die einzelnen Steuerzeichen (Control-Codes) beschrieben. Eine Beschreibung der Steuerzeichen-Folgen (Sequenzen) finden Sie im nächsten Kapitel.

Um die entsprechende Steuerfunktion auszulösen, muß der Code des Steuerzeichens vom Mikrocomputer zum Video-Interface übertragen werden. Näheres dazu entnehmen Sie bitte dem Kapitel "Einsatz unter dem Betriebssystem CP/M".

Für jedes Steuerzeichen ist im folgenden seine Bezeichnung, sein hexadezimaler Code, sein dezimaler Code und seine Funktion angegeben.

## Summer

**BEL**  $07_H$   $7_D$

Durch die Ausgabe des Steuerzeichens BEL ertönt der im Interface eingebaute Summer für etwa 0,5 Sekunden.

## Cursor nach links (Backspace)

**BS**  $08_H$   $8_D$

Der Cursor wird um eine Spalte nach links bewegt. Befindet er sich am linken Schriftfeld-Rand, so bewegt er sich an das Ende der vorstehenden Zeile. Wenn der Cursor sich in der linken oberen Ecke des Schriftfeldes befindet, hat dieses Steuerzeichen keine Wirkung.

## TVI 950-Mode

## Tabulator

**HT****09<sub>H</sub>****9<sub>D</sub>**

Es werden soviele Leerzeichen ausgegeben, bis der Cursor auf dem nächsten *Tabulatorstop* steht. Tabulatorstops befinden sich in jeder achten Spalte. Steht der Cursor bereits auf oder hinter dem letzten Tabulatorstop einer Zeile, so bewegt er sich an den Anfang der nächsten Zeile. Falls der Cursor in der letzten Zeile stand, wird das Schriftfeld um eine Zeile nach oben geschoben. Der Inhalt der oberen Zeile geht verloren, unten wird eine Leerzeile eingefügt.

Spaltenzählung: 0,1,2,3,4,...

Tab.-Positionen: 0,8,16,24,32,40,48,56,64,72,80,88

## Hinweis:

Wie beim original TVI 950-Terminal beginnt die Spaltenzählung der weiter unten beschriebene Escape-Sequenzen "ESC =" und "ESC ?" bei 0, die Spaltenzählung für die Cursor-Positionsanzeige in der Status-Zeile aber bei 1.

## Cursor nach unten, mit Scrollen (Line Feed, Zeilenvorschub)

**LF****0A<sub>H</sub>****10<sub>I</sub>**

Der Cursor wird - unter Beibehaltung der Spaltenposition - um eine Zeile nach unten bewegt. Wenn der Cursor am unteren Schriftfeld-Rand steht, wird das Schriftfeld um eine Zeile nach oben geschoben. Dabei geht der Inhalt der obersten Zeichen-Zeile verloren. Unten wird eine Leerzeile eingefügt.

## Cursor nach oben

**VT****0B<sub>H</sub>****11<sub>D</sub>**

Der Cursor wird - unter Beibehaltung der Spaltenposition - um eine Zeile nach oben bewegt. Steht der Cursor am oberen Schriftfeld-Rand, wird dieses Steuerzeichen ignoriert.

## Cursor nach rechts

**FF****0C<sub>H</sub>****12<sub>D</sub>**

Dieser Code bewegt den Cursor um eine Position nach rechts. Steht der Cursor am Ende einer Zeile, springt er an den Anfang der nächsten Zeile. Steht der Cursor am Ende der letzten Zeile, wird der Inhalt des Schriftfeldes nach oben geschoben (gescrollt). Der Inhalt der oberen Zeile geht dabei verloren, unten wird eine Leerzeile eingefügt.

TVI 950-Mode**Cursor an den Zeilenanfang (Carriage Return, Wagenrücklauf)****CR****0D<sub>H</sub>****13<sub>D</sub>**

Der Cursor wird an den Zeilenanfang bewegt. Der Zeileninhalt bleibt unverändert.

**Cursor nach unten, ohne Scrollen****SYN****16<sub>H</sub>****22<sub>D</sub>**

Dieser Code bewegt den Cursor - unter Beibehaltung der Spaltenposition - um eine Zeile nach unten. Steht der Cursor in der letzten Zeile, wird dieser Code ignoriert.

**Bildschirm löschen, Cursor in Home-Position****SUB****1A<sub>H</sub>****26<sub>D</sub>**

Das komplette Schriftfeld wird mit Leerzeichen gefüllt (gelöscht). Der Cursor wird in die linke obere Ecke des Schriftfeldes (Home-Position) gestellt.

**Cursor in die Home-Position****RS****1E<sub>H</sub>****30<sub>D</sub>**

Dieser Code bewegt den Cursor in die linke obere Ecke des Schriftfeldes (Home-Position). Der Inhalt des Schriftfeldes bleibt unverändert.

**3.2.2 Die Steuer-Sequenzen im TVI 950-Mode**

Im TVI 950-Mode kennt das Interface zusätzlich zu den 1-Byte-Steuerzeichen auch noch Steuerzeichen-Folgen (Sequenzen). Diese Sequenzen beginnen entweder mit dem Zeichen ESC (Escape, 1B<sub>H</sub>) oder mit dem Zeichen DLE (Data Link Escape, 10<sub>H</sub>). Man spricht hier auch von Escape- bzw. DLE-Sequenzen. Um die jeweilige Steuerfunktion auszulösen, müssen alle Bytes einer Steuersequenz vom Mikrocomputer zum Interface übertragen werden.

Für jede Sequenz ist im folgenden ihre Bezeichnung, ihre hexadezimale Byte-Folge, ihre dezimale Byte-Folge und ihre Funktion angegeben.

Die einzelnen Sequenzen sind nach Anwendungsgebieten geordnet.

## TVI 950-Mode

**3.2.2.1 Bildschirm löschen, Cursor Home (Cursor nach links oben)**

ESC *	$1B_H 2A_H$	$27_D 42_D$
ESC +	$1B_H 2B_H$	$27_D 43_D$
ESC ,	$1B_H 2C_H$	$27_D 44_D$
ESC :	$1B_H 3A_H$	$27_D 58_D$
ESC ;	$1B_H 3B_H$	$27_D 59_D$

Die vorstehenden fünf Escape-Sequenzen unterscheiden sich hinsichtlich ihrer Funktion bei dem hier beschriebenen Interface nicht. Durch alle Escape-Sequenzen wird das Schriftfeld mit Leerzeichen gefüllt (gelöscht) und der Cursor in die obere linke Ecke gestellt.

**3.2.2.2 Escape-Sequenzen zum Editieren des Schriftfeldes****Ein Leerzeichen einfügen**

ESC Q	$1B_H 51_H$	$27_D 81_D$
Mit Hilfe der Escape-Sequenz "ESC Q" kann an der Cursor-Position <u>ein</u> Zeichen in eine Zeile eingefügt werden: Der rechte Zeilenteil (ab Cursor-Position einschließlich bis zum rechten Rand) wird um eine Spalte nach rechts verschoben. In der entstehenden Lücke wird ein Leerzeichen eingefügt. Der Cursor steht auf diesem Leerzeichen. Das Zeichen, daß am rechten Schriftfeldrand stand, geht verloren.		

**Ein Zeichen löschen**

ESC W	$1B_H 57_H$	$27_D 87$
Durch die Übertragung der Escape-Sequenz "ESC W" zum Interface, wird das Zeichen an der augenblicklichen Cursor-Position gelöscht. Alle rechts davon befindlichen Zeichen rücken um eine Spalte nach links. In der letzten Spalte der Zeile wird ein Leerzeichen eingefügt. Die Cursor-Position bleibt unverändert.		

**Eine Zeile einfügen**

ESC E	$1B_H 45_H$	$27_D 69_D$
Alle Zeilen einschließlich der aktuellen Cursor-Zeile abwärts werden um eine Zeile nach unten verschoben. Dadurch entsteht eine Leerzeile, in deren erster Spalte der Cursor steht. Die Zeile, die am unteren Schriftfeldrand steht, fällt fort.		

## TVI 950-Mode

**Eine Zeile löschen**

ESC R

 $1B_H 52_H$  $27_D 82_D$ 

Dieser Befehl löscht die aktuelle Cursor-Zeile und rückt alle weiteren Zeilen von unten auf. Am unteren Schriftfeldrand wird eine Leerzeile eingefügt. Der Cursor behält seine Zeilen-Position bei. Er wird aber an den Anfang dieser Zeile gestellt.

**Zeile ab Cursor-Position löschen**

ESC T

 $1B_H 54_H$  $27_D 84_D$ 

ESC t

 $1B_H 74_H$  $27_D 116_D$ 

Durch jeweils eine der beiden oben aufgeführten Escape-Sequenzen wird die Zeile ab der Cursor-Position (einschließlich) bis zum Zeilenende gelöscht. Der restliche Schriftfeld-Inhalt und die Cursor-Position bleiben unverändert.

**Schriftfeld ab Cursor-Position löschen**

ESC Y

 $1B_H 59_H$  $27_D 89_D$ 

ESC y

 $1B_H 79_H$  $27_D 121_D$ 

Durch jeweils eine der beiden oben aufgeführten Escape-Sequenzen wird das Schriftfeld ab der Cursor-Position (einschließlich) bis zur unteren rechten Ecke gelöscht. Die Cursor-Position ändert sich nicht.

**3.2.2.3 Cursor-Positionierung****Cursor-Position setzen**

**ESC = Reihe+20<sub>H</sub> Spalte+20<sub>H</sub> 1B<sub>H</sub> 3D<sub>H</sub> Reihe+20<sub>H</sub> Spalte+20<sub>H</sub> 27<sub>D</sub> 61<sub>D</sub> Reihe+32<sub>D</sub> Spalte+32<sub>D</sub>**  
 Mit Hilfe dieser Escape-Sequenz kann der Cursor auf eine beliebige Position des Schriftfeldes gesetzt werden. Die für diese Escape-Sequenz gültige Reihen- und Spaltenzählung beginnt jeweils bei 0 (siehe auch unten: "Hinweis"). Deshalb hat die linke obere Ecke die Koordinaten 0,0 (Reihe, Spalte). Die Maximalwerte für Reihen- und Spaltenposition hängen von der eingestellten Reihen- (Zeilen-) und Spaltenanzahl ab.

Wertebereiche:

Zeilenanzahl	zul. Reihenposition
22	0...21
24	0...23
26	0...25
28	0...27

## TVI 950-Mode

Spaltenanzahl	zul. Spaltenposition
72	0...71
80	0...79
88	0...87
96	0...95

Werden zu große Werte für Reihe oder Spalte übertragen, so bewegt sich der Cursor zum unteren bzw. zum rechten Schriftfeldrand.

Wie allgemein üblich, wird auch bei diesem Interface die Reihen- und Spaltenposition nicht direkt angegeben. Stattdessen wird auf die Positionsangaben jeweils ein Offset von  $20_H$  ( $32_D$ ) aufaddiert. So werden als Positionsangaben keine Steuerzeichen übertragen.

Beispiel:

Der Cursor soll in die dritte Reihe der ersten Spalte gesetzt werden. Da die Reihen und Spalten jeweils ab Null gezählt werden, ist dies die Position 2,0. Zu diesen Werten muß noch jeweils der Offset  $20_H$  addiert werden. Dies ergibt:  $22_H, 20_H$ . Die komplette Steuerzeichenfolge, die zur Positionierung des Cursors erforderlich ist, lautet also:  $1B_H 3D_H 22_H 20_H$ .

Hinweis:

Wie beim original TVI 950-Terminal beginnt die Spalten- und Reihenzählung zur Positionierung des Cursors durch "ESC =" bei 0, die Zählung für die Cursor-Positionsanzeige in der Status-Zeile aber bei 1.

### Cursor-Position abfragen

ESC ?

$1B_H 3F_H$

$27_D 63_D$

Diese Escape-Sequenz ermöglicht es dem Anwender, die aktuelle Cursor-Position abzufragen. Die für diese Escape-Sequenz gültige Reihen- und Spaltenzählung beginnt jeweils bei 0 (siehe auch unten: "Hinweis"). Das Video-Interface sendet die Reihen- und Spaltenposition jeweils mit einem Offset von  $20_H$  an den Mikrocomputer. Das letzte Zeichen ist immer  $0D_H$  (Carriage Return).

**TVI 950-Mode**

Beispiel:

Der Cursor befindet sich in der dritten Reihe der ersten Spalte. Da die Reihen und Spalten jeweils ab Null gezählt werden, ist dies die Position 2,0. Überträgt der Anwender nun die Sequenz "ESC ?" zum Video-Interface, so sendet dieses die aktuelle Cursor-Position als "Antwort" zum Mikrocomputer. Bei der angenommenen Position würden folgende Bytes als "Antwort" gesendet:

$22_{H}$   $20_{H}$   $0D_{H}$

Das erste Byte ist die Reihen-Position +  $20_{H}$ .

Das zweite Byte ist die Spalten-Position +  $20_{H}$ .

Das dritte Byte ist immer  $0D_{H}$ .

Hinweis:

Wie beim original TVI 950-Terminal beginnt die Spalten- und Reihenzählung zur Abfrage der Cursor-Position mit Hilfe von "ESC ?" bei 0, die Zählung für die Cursor-Positionsanzeige in der Status-Zeile aber bei 1.

**Cursor nach links zum nächsten Tabulator (Backtab)**

**ESC I**

$1B_{H}$   $49_{H}$

$27_{D}$   $73_{D}$

Mit Hilfe dieser Escape-Sequenz wird der Cursor zurück (nach links) zum nächsten Tabulator bewegt. Dies wird auch als "Backtab-Funktion" bezeichnet. Befindet sich der Cursor am Zeilenanfang, wird er auf den letzten Tabulator der vorigen Zeile gesetzt. Befindet er sich in der linken oberen Ecke (Home-Position), wird der Befehl ignoriert.

**3.2.2.4 Zeichensätze, Zeichendarstellung (Attribute), Cursor-Darstellung****Zeichensatz wählen**

**ESC z Auswahlcode**

$1B_{H}$   $7A_{H}$  Auswahlcode

$27_{D}$   $122_{D}$  Auswahlcode

Mit der hier aufgeführten Escape-Sequenz kann einer von acht länderspezifischen Zeichensätzen gewählt werden. Diese Zeichensätze unterscheiden sich nur bei den Zeichencodes  $23_{H}$ ,  $24_{H}$ ,  $40_{H}$ ,  $5B_{H}$ ,  $5C_{H}$ ,  $5D_{H}$ ,  $5E_{H}$ ,  $60_{H}$ ,  $7B_{H}$ ,  $7C_{H}$ ,  $7D_{H}$  und  $7E_{H}$ . Die den einzelnen Zeichensätzen zugeordneten Zeichen können dem Anhang entnommen werden.

Um einen bestimmten Zeichensatz zu wählen, müssen drei Bytes zum Video-Interface übertragen werden:

1. ESC ( $1B_{H}$ ,  $27_{D}$ )
2. z ( $7A_{H}$ ,  $122_{D}$ )
3. Auswahlcode (siehe Tabelle)

## TVI 950-Mode

Der Auswahlcode bestimmt den gewählten Zeichensatz. Die folgende Tabelle zeigt die Zuordnung:

Zeichensatz	Auswahlcode		
	ASCII	hex.	dez.
USA	0	30	48
Frankreich	1	31	49
Deutschland	2	32	50
England	3	33	51
Dänemark	4	34	52
Schweden	5	35	53
Italien	6	36	54
Spanien	7	37	55

Der neue Zeichensatz wirkt sich nicht auf die Zeichen aus, die bereits vor der Zeichensatz-Umschaltung auf dem Bildschirm dargestellt wurden. Er gilt nur für die Zeichen, die nach der Wahl des neuen Zeichensatzes zum Interface übertragen werden.

Auf der Tastatur befinden sich nur die Symbole des USA-Zeichensatzes. Die nachfolgende Tabelle zeigt, welche Tasten zu betätigen sind, um die entsprechenden länderspezifischen Zeichen einzugeben:

Taste	Länderzeichensätze							
	FRA	GER	ENG	DEN	SWE	ITA	SPA	
#	#	#	£	#	#	#	#	
\$	\$	\$	\$	\$	¤	\$	\$	
@	à	§	@	@	@	@	@	
[	°	Ä	[	Æ	Ä	°	í	
\	ç	Ö	\	ø	ö	\	ñ	
]	§	Ü	]	Å	Å	é	]	
^	^	^	^	^	Ü	^	^	
,	,	,	,	,	é	ù	,	
{	é	ä	{	æ	ä	à	"	
	ù	ö		ø	ö	ö	ñ	
}	è	ü	}	å	å	è	)	
-	-	ß	-	-	ü	i	-	

## TVI 950-Mode

## Attribute setzen/löschen

**ESC G Attribute+30<sub>H</sub>****1B<sub>H</sub> 47<sub>H</sub> Attribute+30<sub>H</sub>****27<sub>D</sub> 71<sub>D</sub> Attribute+48<sub>D</sub>**

Die auf dem Bildschirm darzustellenden Zeichen können auf verschiedene Arten angezeigt werden. Soll von der Normaldarstellung abgewichen werden, sind entsprechende Attribute einzuschalten. Folgende Attribute sind möglich:

- unsichtbar
- blinkend
- invers
- unterstrichen
- doppelt breit
- doppelt hoch

Dabei können fast alle Attribute miteinander kombiniert werden. Lediglich bei den Attributen "unsichtbar" und "doppelt hoch" ist eine Kombination miteinander nicht möglich.

Um Attribute ein- oder auszuschalten müssen jeweils drei Bytes zum Interface übertragen werden:

1. ESC (1B<sub>H</sub>, 27<sub>D</sub>)
2. G (47<sub>H</sub>, 71<sub>D</sub>)
3. Attribute + 30<sub>H</sub> (s.u.)

Das dritte Byte "Attribute + 30<sub>H</sub>" gibt die gewünschten Zustände (Ein/Aus) der Attribute an. Es können beliebig viele Attribute gleichzeitig ein- bzw. ausgeschaltet werden.

Eine Änderung der Attribut-Zustände wirkt sich auf alle Zeichen aus, die nach der Änderung zum Interface übertragen werden.

Das Byte "Attribute + 30<sub>H</sub>" ergibt sich durch das Setzen bzw. Löschen bestimmter Bits (zum Ein- bzw. Ausschalten bestimmter Attribute) und durch die anschließende Addition von 30<sub>H</sub>. Welche Bits zum Einschalten bestimmter Attribute gesetzt werden müssen, kann der folgenden Aufstellung entnommen werden:

Attribut	Bit
unsichtbar	0
blinkend	1
invers	2
unterstrichen	3
doppelt breit	4
doppelt hoch	5

Alle anderen Bits sind reserviert und dürfen nicht gesetzt werden.

## TVI 950-Mode

Beispiel:

Die nachfolgenden Zeichen sollen blinkend und unterstrichen dargestellt werden. Dazu sind die Attribute "blinkend" und "unterstrichen" ein- und alle anderen Attribute auszuschalten. Es ergibt sich das folgende Bitmuster:

Bit-Nummer	7	6	5	4	3	2	1	0
Bitmuster	0	0	0	0	1	0	1	0
0					A			

Aus dem Bitmuster ergibt sich der Wert  $0A_H$ . Zu diesem Wert muß nun noch der Offset  $30_H$  addiert werden. Man erhält dann den Wert  $3A_H$ . Die vollständige Sequenz, die zum Einschalten der gewünschten Attribute zum Interface übertragen werden muß, lautet also:  $1B_H\ 47_H\ 3A_H$ . Durch die Übertragung dieser Byte-Folge werden die Attribute "unsichtbar", "invers", "doppelt breit" und "doppelt hoch" ausgeschaltet, da die entsprechenden Bits im Bitmuster "0" sind.

Im Anhang finden Sie eine Tabelle, die Ihnen die Ermittlung des Bytes "Attribute +  $30_H$ " erleichtert.

Mit der Escape-Sequenz "ESC G 0" ( $1B_H\ 47_H\ 30_H$ ) können alle Attribute gleichzeitig ausgeschaltet werden.

#### Besondere Hinweise:

Bei der Verwendung der Attribute "doppelt hoch" und "doppelt breit" ist zu beachten, daß sich die meisten Steuerbefehle auf Zeichen einfacher Höhe und Breite beziehen. Bei Zeichen mit doppelter Höhe und/oder Breite können sie entweder keine oder aber eine andere Wirkung als vorgesehen haben. Der einzige Steuercode, der die Schriftgröße berücksichtigt, ist "Line Feed" (Zeilenvorschub,  $0A_H$ ). Falls das Attribut "doppelt hoch" aktiv ist, werden zwei Standard-Zeilenvorschübe ausgeführt. Dadurch entsteht bei Abschluß jeder Zeile mit CR/LF ein brauchbares Schriftbild. Bei Zeilenüberlauf wird auch bei doppelter Zeichenhöhe nur ein Zeilenvorschub ausgeführt, wodurch die untere Hälfte der vorigen Zeile von der nächsten überschrieben wird.

Bei doppelt hoher Schrift werden die einzelnen Zeichen grundsätzlich nach oben erweitert. Sie überschreiben dabei eventuell Zeichen, die in der darüberliegenden Zeile stehen. Steht der Cursor in der obersten Zeile (z.B. Home-Position: 0,0) so wird der Cursor nach Übertragung des ersten Zeichens aber vor der Zeichendarstellung vom Video-Interface eine Zeile tiefer positioniert. Dadurch lassen sich auch nach einem Cursor-Home-Befehl sofort doppelt hohe Zeichen darstellen.

Bei doppelt breiter Schrift kann es zu Verzerrungen kommen, wenn doppelt breite Schrift auf einer ungeraden Spalte beginnt (Zählung ab 0). In diesem Fall werden beim Zeilenüberlauf in der nächsten Zeile jeweils nicht zusammenhängende Buchstabenhälften angezeigt.

## TVI 950-Mode

**Halbe Helligkeit****ESC )****1B<sub>H</sub> 29<sub>H</sub>****27<sub>D</sub> 41<sub>D</sub>**

Das Interface kann die Zeichen in zwei Helligkeitsstufen anzeigen. Mit der hier aufgeführten Escape-Sequenz wird für alle nachfolgend übertragenen Zeichen die halbe Helligkeit gewählt. Normaleinstellung ist die volle Helligkeit. "ESC (" schaltet wieder auf die volle Helligkeit zurück.

**Volle Helligkeit****ESC (****1B<sub>H</sub> 28<sub>H</sub>****27<sub>D</sub> 40<sub>D</sub>**

Mit der hier aufgeführten Escape-Sequenz kann die Zeichendarstellung auf die volle Helligkeit (Normaleinstellung) geschaltet werden. Siehe auch "ESC )".

**Bildhintergrund hell (bright)****Normaldarstellung als "dunkle Zeichen auf hellem Grund"****ESC b****1B<sub>H</sub> 62<sub>H</sub>****27<sub>D</sub> 98<sub>D</sub>**

Mit dieser Escape-Sequenz wird für alle Zeichen ohne das Attribut "invers" die Darstellungsart "dunkle Zeichen auf hellem Grund" gewählt. Für Zeichen mit dem Attribut "invers" gilt: helle Zeichen auf dunklem Grund. Dies ist die Grundeinstellung, wenn über den Schalter S1.4 ein heller Bildhintergrund gewählt wurde. Diese Escape-Sequenz wirkt sich auf alle Zeichen aus (auch auf die bereits angezeigten). Siehe auch "ESC d".

**Bildhintergrund dunkel (dark)****Normaldarstellung als "helle Zeichen auf dunklem Grund"****ESC d****1B<sub>H</sub> 64<sub>H</sub>****27<sub>D</sub> 100<sub>D</sub>**

Mit dieser Escape-Sequenz wird für alle Zeichen ohne das Attribut "invers" die Darstellungsart "helle Zeichen auf dunklem Grund" gewählt. Für Zeichen mit dem Attribut "invers" gilt: dunkle Zeichen auf hellem Grund. Dies ist die Grundeinstellung, wenn über den Schalter S1.4 ein dunkler Bildhintergrund gewählt wurde. Diese Escape-Sequenz wirkt sich auf alle Zeichen aus (auch auf die bereits angezeigten). Siehe auch: "ESC b".

**Control-Mode einschalten****ESC U****1B<sub>H</sub> 55<sub>H</sub>****27<sub>D</sub> 85<sub>D</sub>**

Mit dieser Escape-Sequenz wird der "Control-Mode" eingeschaltet. In diesem Mode werden alle Steuerzeichen, die das Video-Interface empfängt, nicht ausgeführt, sondern als Control-Zeichen angezeigt. Siehe auch das Kapitel "Der Control-Mode".

## TVI 950-Mode

Beispiel:

Wird das Steuerzeichen BEL ( $07_H$ ) bei eingeschaltetem Control-Mode vom Mikrocomputer zum Interface übertragen, so ertönt nicht der Summer des Interfaces. Stattdessen wird an der Cursor-Position ein inverses "G" mit halber Helligkeit dargestellt. Das "G" steht für die Tastenkombination "Control G", die den Code  $07_H$  erzeugt.

Hinweis:

Da alle Steuerzeichen nur noch angezeigt aber nicht mehr ausgeführt werden, kann der Control-Mode nur über das Set-Up-Menü (Control Mode AUS) verlassen werden. Siehe Kapitel "Das Set-Up-Menü".

## Cursor-Darstellung

**ESC . Darstellung**

**$1B_H \ 2E_H$  Darstellung**

**$27_D \ 46_D$  Darstellung**

Mit Hilfe dieser Escape-Sequenz kann die Art der Cursor-Darstellung den Wünschen des Anwenders angepaßt werden. Hierzu sind drei Bytes zum Interface zu übertragen:

1. ESC ( $1B_H$ ,  $27_D$ )
2. . ( $2E_H$ ,  $46_D$ )
3. Darstellung (siehe Tabelle)

Das dritte Byte ("Darstellung") bestimmt die Darstellungsart des Cursors. Welches "Darstellung"-Byte zum Interface übertragen werden muß, kann der folgenden Tabelle entnommen werden:

Darstellungsart	"Darstellung"		
	ASCII	hex.	dez.
keinen Cursor anzeigen	0	$30_H$	$48_D$
blinkender Block	1	$31_H$	$49_D$
stehender Block	2	$32_H$	$50_D$
blinkender Unterstrich	3	$33_H$	$51_D$
stehender Unterstrich	4	$34_H$	$52_D$

Beispiel:

Soll für den Cursor die Darstellungsart "blinkender Unterstrich" gewählt werden, so ist die Byte-Folge  $1B_H \ 2E_H \ 33_H$  zum Interface zu übertragen.

**TVI 950-Mode****3.2.2.5 Die DLE-Sequenzen im TVI 950-Mode**

Die DLE-Sequenzen sind identisch mit den DLE-Sequenzen im MAT 85-Mode.

**TVI 950-Mode einschalten****DLE DLE 1** $10_{\text{H}} 10_{\text{H}} 31_{\text{H}}$  $16_{\text{D}} 16_{\text{D}} 49_{\text{D}}$ 

Mit dieser Sequenz kann in den TVI 950-Mode umgeschaltet werden. Befindet man sich bereits im TVI 950-Mode, so wird diese Sequenz ignoriert.

**MAT 85-Mode einschalten****DLE DLE 2** $10_{\text{H}} 10_{\text{H}} 32_{\text{H}}$  $16_{\text{D}} 16_{\text{D}} 50_{\text{D}}$ 

Mit dieser Sequenz kann in den MAT 85-Mode geschaltet werden. Nach der Umschaltung gelten die Steuerzeichen-Definitionen des MAT 85-Modes.

**Aktuellen Mode abfragen****DLE DLE ?** $10_{\text{H}} 10_{\text{H}} 3F_{\text{H}}$  $16_{\text{D}} 16_{\text{D}} 63_{\text{D}}$ 

Mit Hilfe dieser Sequenz kann der Anwender abfragen, welcher Mode (MAT 85 oder TVI 950) momentan aktiv ist. Nach dem Empfang dieser Sequenz "antwortet" das Interface mit drei Bytes:

Antwort-Bytes	aktiver Mode
$4D_{\text{H}} \quad 31_{\text{H}} \quad 0D_{\text{H}}$	TVI 950-Mode
$4D_{\text{H}} \quad 32_{\text{H}} \quad 0D_{\text{H}}$	MAT 85-Mode

Beachten Sie hierzu bitte auch das Kapitel "Einsatz unter dem Betriebssystem CP/M".

**Software-Reset****DLE DLE @** $10_{\text{H}} 10_{\text{H}} 40_{\text{H}}$  $16_{\text{D}} 16_{\text{D}} 64_{\text{D}}$ 

Durch diese Sequenz kann ein sogenannter Software-Reset des Video-Interfaces ausgelöst werden. Das heißt, das Interface wird so eingestellt, wie es nach dem Anlegen der Betriebsspannung eingestellt sein würde.

## TVI 950-Mode

**Abfrage der Software-Version****DLE DLE V** $10_{\text{H}}$   $10_{\text{H}}$   $56_{\text{H}}$  $16_{\text{D}}$   $16_{\text{D}}$   $86_{\text{D}}$ 

Mit Hilfe dieser Sequenz kann der Anwender die Versionsnummer der Software im EPROM der Interface-Baugruppe abfragen. Die Version ist durch zwei Ziffern gekennzeichnet (Versions-Nummer). Nach dem Empfang dieser Sequenz "antwortet" das Interface mit fünf Bytes:

Beispiel für eine Antwort bei Version 1.0:

$56_{\text{H}}$      $31_{\text{H}}$      $2F_{\text{H}}$      $30_{\text{H}}$      $0D_{\text{H}}$

Das erste Byte der Antwort ist immer  $56_{\text{H}}$  (ASCII "V").

Das zweite Byte entspricht dem ASCII-Code der ersten Ziffer der Versions-Nummer (hier "1").

Das dritte Byte ist (bei dieser Interface-Hardware) immer  $2F_{\text{H}}$ .

Das vierte Byte entspricht dem ASCII-Code der zweiten Ziffer der Versions-Nummer (hier "0").

Das fünfte Byte der Antwort ist immer  $0D_{\text{H}}$ .

Beachten Sie hierzu bitte auch das Kapitel "Einsatz unter dem Betriebssystem CP/M".

## Einsatz unter MAT 85

### 4 Einsatz unter dem Betriebsprogramm MAT 85

Für den Einsatz unter dem Betriebsprogramm MAT 85 des MFA-Mikrocomputers ist der MAT 85-Mode des Video-Interfaces vorgesehen. Der Datenaustausch zwischen CPU-Baugruppe und Video-Interface erfolgt über die Systembus-Leitungen Tx und Rx (siehe Bild 7).

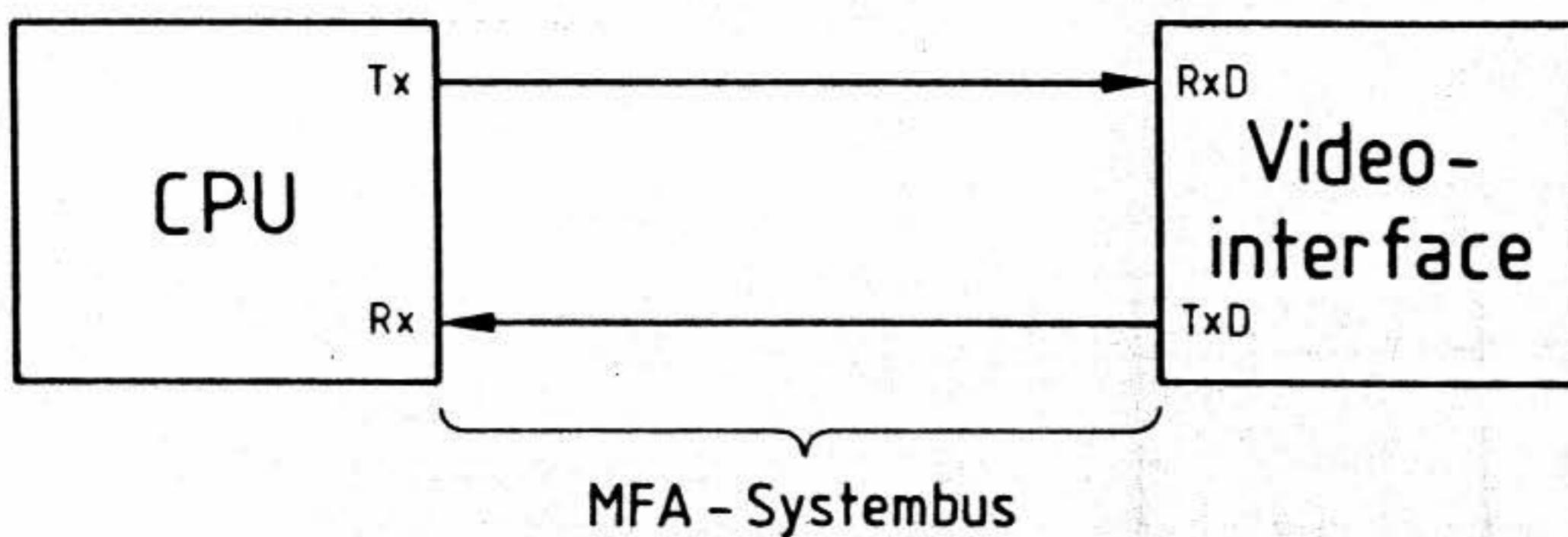


Bild 7: Datenaustausch zwischen CPU und Video-Interface (MAT 85)

Vor der Inbetriebnahme sind die Schalter der Interface-Baugruppe entsprechend dem nachfolgenden Kapitel "Schalterstellungen" einzustellen.

Die Kapitel "Nutzung in Assembler Programmen" und "Nutzung unter BFZ-STEUER-BASIC" enthalten Informationen für Anwender, welche die Steuerfunktionen des Video-Interfaces in eigenen Assembler- bzw. BASIC-Programmen nutzen wollen.

An dieser Stelle soll darauf hingewiesen werden, daß sowohl der MAT 85-Assembler als auch das BFZ-STEUER-BASIC bei der Eingabe von Programmzeilen alle Kleinbuchstaben in Großbuchstaben umwandeln. So können diese Programme die eingegebenen Assembler- bzw. BASIC-Anweisungen einfacher erkennen.

Beachten Sie bitte, daß nach dem Einschalten des Mikrocomputer-Systems die Leertaste betätigt werden muß. Dies gilt auch dann, wenn auf dem Bildschirm keine Zeichen angezeigt werden.

Einsatz unter MAT 85**4.1 Schalterstellungen**

Auf der Interface-Baugruppe befinden sich drei Schalter-Reihen (siehe auch Bestückungsplan im Anhang) zur Grundeinstellung des Video-Interfaces beim Einschalten. Jede der mit S1, S2 und S3 bezeichneten Schalter-Reihen besteht aus acht Schaltern. S1.1 bezeichnet den Schalter Nummer 1 in der Schalter-Reihe S1. Die anderen Schalter sind entsprechend nummeriert. Für das Betriebssystem MAT 85 sind die Schalter folgendermaßen einzustellen:

**- Schalter-Reihe S1:**

Schalter	Stellung	eingestellte Funktion
S1.1	ON	MAT 85-Mode
S1.2	OFF	blinkender Cursor
S1.3	OFF	Strich-Cursor
S1.4	OFF	dunkler Bildhintergrund
S1.5	OFF	reserviert, immer OFF!
S1.6	OFF	DTR-Handshake
S1.7	OFF	reserviert, immer OFF!
S1.8	OFF	Statuszeile aus

**- Schalter-Reihe S2**

Schalter	Stellung	eingestellte Funktion
S2.1	OFF	Wortlänge 7 Bit
S2.2	OFF	Zeichensatz USA <sup>1)</sup>
S2.3	OFF	dto. <sup>1)</sup>
S2.4	OFF	dto. <sup>1)</sup>
S2.5	ON	Zeilenanzahl 24
S2.6	OFF	dto.
S2.7	ON	Spaltenanzahl 80
S2.8	OFF	dto.

<sup>1)</sup> Entspricht dem Zeichensatz des Video-Interfaces BFZ/MFA 8.2

## Einsatz unter MAT 85

## - Schalter-Reihe S3

Schalter	Stellung	eingestellte Funktion
S3.1	ON	2 Stopbits
S3.2	ON	Mark Parity
S3.3	OFF	dto.
S3.4	ON	dto.
S3.5	ON	1200 Baud
S3.6	OFF	dto.
S3.7	ON	dto.
S3.8	OFF	dto.

## Anmerkung:

Die Schalter S1.2 (Cursor-Art), S1.3 (Cursor-Form), S1.4 (Bildhintergrund), S1.8 (Statuszeile) und S2.2, S2.3, S2.4 (alle drei zur Einstellung des Zeichensatzes) können - den persönlichen Wünschen entsprechend - anders eingestellt werden. Eine andere Einstellung dieser Schalter beeinträchtigt die Funktion nicht. Angaben über die durch diese Schalter einstellbaren Optionen entnehmen Sie bitte dem Anhang.

## 4.2 Nutzung in Assembler-Programmen

## 4.2.1 Zeichenausgabe

Da unter dem Betriebsprogramm MAT 85 der MAT 85-Mode des Interfaces genutzt wird, können grundsätzlich alle Ausgabe-Routinen, die in der MAT 85-Beschreibung dokumentiert sind, verwendet werden. Hier sollen nur einige dieser Routinen genannt werden:

## WCHAR, Adresse 0052

Sendet das Zeichen, dessen Code sich im Akkumulator befindet, zum Interface

```

MVI A,0D      ;CODE FUER WAGENRUECKLAUF
CALL 0052     ;CODE AUSGEBEN
MVI A,'X'     ;CODE FUER 'X'
CALL 0052     ;CODE AUSGEBEN
JMP  0040     ;ENDE, ZURUECK ZUR MAT 85-KOMMANDOEINGABE

```

---

Einsatz unter MAT 85

---

**WCHARI, Adresse 0055**

Gibt 1 Zeichen, dessen Code nach dem CALL-Befehl im Speicher steht, aus.

```
CALL 0055 ;NACHFOLGENDEN CODE AUSGEBEN
DB    0D    ;CODE FUER WAGENRUECKLAUF
CALL 0055 ;NACHFOLGENDEN CODE AUSGEBEN
DB    'X'   ;CODE FUER 'X'
JMP   0040 ;ENDE, ZURUECK ZUR MAT 85-KOMMANDOEINGABE
```

**WBUFI, Adresse 006D**

Gibt den hinter dem CALL-Befehl stehenden Text auf dem Bildschirm aus. Am Ende des Textes muß als Endemarke das Byte 00<sub>H</sub> stehen.

```
CALL 006D ;NACHFOLGENDE CODES AUSGEBEN
DB    0D, 'X';CODES FUER WAGENRUECKLAUF UND 'X'
DB    00    ;TEXT-ENDEMARKE
JMP   0040 ;ENDE, ZURUECK ZUR MAT 85-KOMMANDOEINGABE
```

**WBUF, Adresse 0BA1**

Gibt Text aus einem Textpuffer aus, dessen Anfangsadresse sich vor dem Aufruf der Routine im HL-Registerpaar befinden muß. Am Ende des Textes muß als Endemarke 00<sub>H</sub> stehen.

```
LXI H, BUF ;ANFANGSADRESSE DES PUFFERS
CALL 0BA1 ;CODES AUSGEBEN
JMP   0040 ;ENDE, ZURUECK ZUR MAT 85-KOMMANDOEINGABE
BUF:   DB    0D, 'X';AUSZUGEBENDE CODES
       DB    00    ;TEXT-ENDEMARKE
```

#### 4.2.2 Zeicheneingabe

Alle vom Interface gesendeten Zeichen - gleichgültig, ob sie über die Tastatur eingegeben wurden oder die "Antwort" einer DLE-Sequenz sind - können unter Verwendung des MAT 85-Unterprogramms "RCHAR" (Adresse 0043<sub>H</sub>) eingelesen werden.

Beispiel:

Das nachstehende Programm gibt im ersten Teil die Steuersequenz "DLE DLE ?" zur Abfrage des aktuellen Modes aus. Die aus 3 Bytes bestehende Antwort des Terminals wird in einen Puffer eingelesen und anschließend angezeigt. Die Anzeige kann nicht sofort nach dem Empfang eines Antwort-Bytes erfolgen, da sonst die folgenden Bytes verloren gehen.

## Einsatz unter MAT 85

Im zweiten Teil des Programms werden in einer Endlosschleife Zeichen vom Video-Interface gelesen und sofort zum Video-Interface zurückgesendet. Dies gilt auch für Steuerzeichen. Durch die Betätigung der CR-Taste (Code  $0D_H$ ) wird das Programm beendet.

Begründet durch die verwendete Routine "RCHAR" des MAT 85-Betriebsprogramms, kann das Programm auch durch die Betätigung der ESC-Taste abgebrochen werden.

```

START:    LXI H, MODE      ;ZEIGER AUF STEUERZ. FUER MODEABFRAGE
          CALL 0BA1       ;GEBE STEUERZEICHEN AUS
          LXI H, BUF       ;PUFFER FUER DIE ANTWORT-ZEICHEN
          MVI B, 3        ;ZAEHLER: DREI ZEICHEN EINLESEN
LOOP:     CALL 0043       ;LESE ZEICHEN IN DEN AKKUMULATOR
          MOV M,A        ;SCHREIBE ZEICHEN IN DEN PUFFER
          INX H          ;ZEIGER WEITERSTELLEN
          DCR B          ;ANTWORT KOMPLETT GELESEN ?
          JNZ LOOP       ;NEIN --> LOOP
          LXI H, BUF       ;ZEIGER AUF 1. ANTWORT-ZEICHEN
          CALL 0BA1       ;ANTWORT AUF DEM BILDSCHIRM ANZEIGEN
;
TEIL2:    CALL 0043       ;LESE ZEICHEN IN DEN AKKUMULATOR
          CPI 0D          ;CARRIAGE-RETURN ?
          JZ 0040         ;JA --> 0040 (ENDE)
          CALL 0052       ;NEIN (NICHT 0D) --> ZEIGE ZEICHEN AN
          JMP TEIL2       ;LESE WEITERE ZEICHEN
;
MODE:    DB 10,10,'?' ;STEUERZEICHEN F. MODEABFRAGE
          DB 00           ;ENDEMARKE
BUF:     DB 00,00,00   ;BUFFER FUER ANTWORT
          DB 00           ;ENDEMARKE

```

## 4.2.3 Ein Programmbeispiel in Assembler-Sprache

Das nachstehende Programm fordert den Anwender auf, eine Ziffer einzugeben. Der Aufforderungstext wird mit Hilfe des Unterprogramms "TXTAUS" (MAT 85-Routine  $0BA1_H$ ) ausgegeben. Über die Routine  $0043_H$  wird die Tastatur abgefragt. Das Programm überprüft, ob tatsächlich eine Ziffer (ASCII-Code  $30_H$  bis  $39_H$ ) eingegeben wurde. Ist dies der Fall, gibt das Programm die entsprechende Ziffer als Wort auf dem Bildschirm aus. Wurde keine Ziffer eingegeben, wird die Meldung "KEINE ZIFFER" angezeigt. Das hier beschriebene Programm wertet die Betätigung der "CR"-Taste als Abbruchwunsch aus. Ein Abbruch ist außerdem durch Betätigung der "ESC"-Taste möglich. Dies ergibt sich durch die verwendete MAT 85-Routine  $0043_H$ .

## Einsatz unter MAT 85

```

START:    LXI H,TEXT1      ;ZEIGER AUF 1. TEXT
          CALL 0BA1       ;TEXT AUSGEBEN
          CALL 0043       ;ZEICHEN VON TASTATUR EINLESEN
          CPI 0D          ;CR-TASTE BETAETIGT ?
          JZ 0040         ;WENN JA --> 0040H, ENDE
          CPI 30          ;ASCII-CODE < 30H ?
          JC NOZIFF       ;WENN JA --> NOZIFF
          CPI 3A          ;ASCII-CODE >= 3AH ?
          JNC NOZIFF      ;WENN JA --> NOZIFF
          SUI 2F          ;ASCII-CODE AUF 1...10 (DEZ) REDUZIEREN
          MOV B,A          ;ERMITTELTN WERT (ZAEHLER) NACH B
          LXI H,ZIFF       ;ZEIGER AUF ZIFFERN-TEXTE
SUCHO:    MOV A,M          ;ZEICHEN LESEN
          INX H            ;ZEIGER AUF NAECHSTES ZEICHEN STELLEN
          ORA A            ;00H (TRENNZEICHEN) ?
          JNZ SUCHO        ;WENN NEIN --> SUCHO
          DCR B            ;ZAEHLER MINUS EINS
          JNZ SUCHO        ;ZAEHLER NICHT NULL --> SUCHO
          CALL 0BA1        ;TEXT AUSGEBEN
          JMP START        ;AUF EIN NEUES
;-----
NOZIFF:   LXI H,NZTXT      ;ZEIGER AUF FEHLERMELDUNG
          CALL 0BA1        ;MELDUNG AUSGEBEN
          JMP START        ;AUF EIN NEUES
;-----
TEXT1:   DB 0D,0A          ;WAGENRUECKLAUF, ZEILENVORSCHUB
          DB 'BITTE EINE ZIFFER EINGEBEN'
          DB 0D,0A          ;WAGENRUECKLAUF, ZEILENVORSCHUB
          DB 00              ;ENDEMARKE
NZTXT:   DB 07              ;BELL
          DB 'KEINE ZIFFER'
          DB 0D,0A          ;WAGENRUCKLAUF, ZEILENVORSCHUB
          DB 00              ;ENDEMARKE
ZIFF:    DB 00              ;MARKE
          DB 'NULL'
          DB 00              ;MARKE
          DB 'EINS'
          DB 00              ;MARKE
          DB 'ZWEI'
          DB 00              ;MARKE
          DB 'DREI'
          DB 00              ;MARKE
          DB 'VIER'

```

---

Einsatz unter MAT 85

---

```
DB    00          ;MARKE
DB    'FUENF'
DB    00          ;MARKE
DB    'SECHS'
DB    00          ;MARKE
DB    'SIEBEN'
DB    00          ;MARKE
DB    'ACHT'
DB    00          ;MARKE
DB    'NEUN'
DB    00          ;MARKE
```

#### 4.3 Nutzung unter BFZ-STEUER-BASIC

##### 4.3.1 Zeichenausgabe

Darstellbare Zeichen (wie Buchstaben und Ziffern) können unter BFZ-STEUER-BASIC mit Hilfe der PRINT-Anweisung ausgegeben werden:

```
10 PRINT "Video-Interface"
```

Für die Ausgabe von Steuerzeichen ist ein kleines Hilfsprogramm erforderlich. Es kann mit dem nachstehenden BASIC-Programm in den RAM-Speicher eingetragen werden:

```
32000 POKE DEC(E000),125
32010 POKE DEC(E001),205
32020 POKE DEC(E002),82
32030 POKE DEC(E003),0
32040 POKE DEC(E004),201
```

Das Hilfsprogramm besteht aus drei Assembler-Anweisungen:

```
E000 MOV A,L ; Lade Ausgabe-Zeichen vom L-Register in den Akku
E001 CALL 0052 ; Gebe Zeichen aus
E004 RET       ; Zurück zum BASIC-Programm
```

Das Hilfsprogramm ermöglicht die Ausgabe aller ASCII-Zeichen. Dazu wird es über die USR-Funktion aufgerufen. Alle vier nachstehend aufgeführten Programm-Zeilen sind gleichwertig und zeigen verschiedene Arten des USR-Aufrufs. In allen vier Fällen wird das Steuerzeichen VT (Cursor Up - Cursor hoch - Code=0B<sub>H</sub> (11 dez.)) ausgegeben. Vergleiche auch: Anleitung zum BFZ-STEUER-BASIC in den Fachtheoretischen Übungen (FTÜ).

## Einsatz unter MAT 85

```
100 X=USR(DEC(E000),DEC(0B)) : REM --- GIBT STEUERZEICHEN OB AUS
110 X=USR(DEC(E000),11) : REM --- WIE OBEN
120 C=DEC(0B) : X=USR(DEC(E000,C)) : REM --- WIE OBEN
130 C=11 : X=USR(DEC(E000,C)) : REM --- WIE OBEN
```

Bitte beachten Sie, daß das Hilfsprogramm vor seinem ersten Aufruf im RAM installiert werden muß! Dazu können die oben aufgeführten BASIC-Programmzeilen 32000 bis 32040 als Unterprogramm in Ihr Programm eingefügt und zu Beginn aufgerufen werden:

```
1 GOSUB 32000 : REM --- HILFSROUTINE IN DAS RAM LADEN
:
:
111 X=USR(DEC(E000),C) : REM --- ZEICHEN (CODE IN C) AUSGEBEN
:
:
31999 END
32000 POKE DEC(E000),125
32010 POKE DEC(E001),205
32020 POKE DEC(E002),82
32030 POKE DEC(E003),0
32040 POKE DEC(E004),201
32050 RETURN
```

#### 4.3.2 Zeicheneingabe

Tastatureingaben können mit Hilfe der INPUT- oder der \$-Anweisung eingelesen werden. Genaueres über diese Anweisungen können Sie der Anleitung zum BFZ-STEUER-BASIC im FTÜ-Band entnehmen.

Beispiele:

```
10 INPUT "GEWICHT (KG) ",G
20 PRINT "DAS GEWICHT BETRAEGT";G;"KILOGRAMM"
30 PRINT "BITTE BETAETIGEN SIE EINE TASTE"
40 T=$
50 PRINT "DIE BETAETIGTE TASTE HAT DEN CODE";T;"(DEZ.)"
```

Die DLE-Sequenzen "DLE DLE ?" (Mode-Abfrage) und "DLE DLE V" (Versions-Abfrage) können im **BFZ-STEUER-BASIC** nicht verwendet werden, da der Basic-Interpreter die "Antwort" des Interfaces nicht entgegennehmen kann. Sollte der TVI 950-Mode verwendet werden, so kann außerdem die Escape-Sequenz "ESC ?" (Abfrage der Cursor-Position) aus dem gleichen Grund nicht genutzt werden.

## Einsatz unter MAT 85

**4.3.3 Ein Programmbeispiel in BFZ-STEUER-BASIC**

Das nachstehende Programm soll die einzelnen Aus- und Eingabemöglichkeiten des BFZ-STEUER-BASICs in Verbindung mit dem Video-Interface darstellen. Texte werden über eine PRINT-Anweisung ausgegeben, Steuerzeichen mit Hilfe eines Maschinensprache-Programms. Dieses Maschinensprache-Programm wird durch die Programmzeilen 32000 bis 32050 ab Adresse E000 in das RAM geladen. Hier soll nochmals darauf hingewiesen werden, daß dieser Ladevorgang vor dem ersten Aufruf des Maschinensprache-Programms erfolgen muß. Der Aufruf des Maschinensprache-Programms erfolgt über die USR-Anweisung in Programmzeile 32100. Die USR-Anweisung wurde als Unterprogramm geschrieben. Der Code des auszugebenden Steuerzeichens muß vor dem Unterprogramm-Aufruf (also vor GOSUB 32100) in die Variable C geladen werden. Zeile 230 zeigt die Anwendung der "\$"-Funktion.

```
1 GOSUB 32000 : REM *** HILFSROUTINE IN DAS RAM LADEN
10 C=7 : REM *** CODE FUER "BEL"
20 GOSUB 32100 : REM *** CODE AUSGEBEN
30 PRINT "1. TEXTZEILE" : REM *** TEXT AUSGEBEN
40 PRINT : REM EINE LEERZEILE ERZEUGEN
50 PRINT "2. TEXTZEILE"
60 REM *** NUN DEN CURSOR ZWISCHEN DIE BEIDEN TEXTZEILEN STELLEN
70 REM *** (DER CURSOR STEHT NACH DER LETZTEN PRINT-ANWEISUNG
80 REM *** UNTER DEM TEXT "2. TEXTZEILE". DAHER MUSS DER CURSOR
90 REM *** UM ZWEI ZEILEN NACH OBEN GESTELLT WERDEN. DER CODE
100 REM *** FUER "CURSOR HOCH" IST 11 (DEZ))
110 C=11 : GOSUB 32100 : GOSUB 32100
120 REM *** ACHTEN SIE BEIM EINGEBEN DIESES PROGRAMMS DARAUF,
130 REM *** DASS AM ENDE DER NACHFOLGENDEN ZEILE EIN ";" STEHT
140 PRINT "-----";
150 REM *** NUN CURSOR UNTER "2. TEXTZEILE" STELLEN
160 C=10 : REM *** CODE FUER "CURSOR ABWAERTS" (LINE FEED)
170 GOSUB 32100 : GOSUB 32100 : REM *** CURSOR 2 ZEILEN TIEFER
180 C=13 : REM *** CODE FUER "WAGENRUECKLAUF" (CARRIAGE RETURN)
190 GOSUB 32100 : REM *** CURSOR AN ZEILENANFANG
200 INPUT "WIE ALT SIND SIE ? ",A
210 PRINT "SIE BEHAUPTEN,";A;"JAHRE ALT ZU SEIN"
220 PRINT "BITTE BETAETIGEN SIE EINE TASTE"
230 C=$ : REM *** TASTATURABFRAGE (VERGL. BASIC-ANLEITUNG)
240 PRINT "DIE VON IHNEN BETAETIGTE TASTE ERZEUGT DEN CODE";
250 PRINT C;"(DEZ) BZW.";#C;" (HEX)" : REM *** CODE ANZEIGEN
260 PRINT "---- ENDE ----"
31999 END
```

## Einsatz unter MAT 85

```
32000 POKE DEC(E000),125
32010 POKE DEC(E001),205
32020 POKE DEC(E002),82
32030 POKE DEC(E003),0
32040 POKE DEC(E004),201
32050 RETURN
32100 X=USR(DEC(E000),C) : RETURN : REM *** ZEICHENAUSGABE
```

## Einsatz unter CP/M

## 5 Einsatz unter dem Betriebssystem CP/M

Zum Betrieb des Video-Interfaces BFZ/MFA 8.4 unter CP/M ist eine Serienschnittstelle erforderlich. Hierzu kann entweder die "V24/RS232-Schnittstelle" BFZ/MFA 4.10 oder die "Programmierbare Serienschnittstelle" BFZ/MFA 4.4 verwendet werden. Bei der "Programmierbaren Serienschnittstelle" sind die V24/TTL- und TTL/V24-Pegewandler nachzurüsten. Dies ist im FPÜ-Band 4 "CP/M-Ausbaustufe", Kapitel "Aufbau und Inbetriebnahme" näher beschrieben. Eine Serienschnittstelle, die nach der Anleitung im FPÜ-Band 4 für das Video-Interface BFZ/MFA 8.2 eingestellt worden ist, muß noch geringfügig modifiziert werden. Die Modifikation ermöglicht einen Handshake-Betrieb.

- Bei der V24/RS232-Schnittstelle ist eine Drahtbrücke nachzurüsten. Diese Schnittstelle kann nach der Umrüstung sowohl mit dem Video-Interface BFZ/MFA 8.4 als auch mit dem Video-Interface BFZ/MFA 8.2 betrieben werden.
- Bei der Programmierbaren Serienschnittstelle BFZ/MFA 4.4 sind zwei Drahtbrücken nachzurüsten und eine andere zu entfernen. Nach der Umrüstung kann diese Schnittstelle nur noch mit dem hier beschriebenen Video-Interface BFZ/MFA 8.4 betrieben werden (nicht mehr mit dem Video-Interface BFZ/MFA 8.2).

Nachstehend eine kurze Aufstellung der zum Betrieb notwendigen Brücken-Einstellungen und der notwendigen Umrüstung.

Programmierbare Serienschnittstelle BFZ/MFA 4.4	V24/RS232-Schnittstelle BFZ/MFA 4.10
<p>Baudatenbrücke C geschlossen (ergibt 1200 Bd)</p> <p>geschlossene Lötbrücken: P2 bis P6</p> <p>geöffnete Lötbrücken: P1</p> <p>Baugruppe mit Pegewandlern nachgerüstet (im FPÜ-Band 4, Übung "Aufbau und Betriebnahme" beschrieben)</p> <p>Umrüstung:</p> <p>1. <u>Brücke zwischen dem Leiterplattenanschluß DSR und 0 V (GND) entfernen</u> ! (falls vorhanden)</p> <p>2. Drahtbrücke von Anschluß 28c der VG64-Steckerleiste nach Pin 13 des IC8 (1488)</p> <p>3. Drahtbrücke von Pin 11 des IC8 (1488) zum Leiterplatten- anschluß DSR</p>	<p>Baudatenbr. D geschl. (ergibt 1200 Bd)</p> <p>geschlossene Lötbrücken: P3, P4, 9, 14, 15, 18</p> <p>geöffnete Lötbrücken: 1 bis 8, 10 bis 13, 16, 17</p> <p>kein Stecker in die 25pol. Buchsenleiste auf der Frontplatte einge- steckt</p> <p>Umrüstung:</p> <p>Drahtbrücke von Anschluß 28c der VG64-Steckerleiste nach Lötbrücke 14</p>

## Einsatz unter CP/M

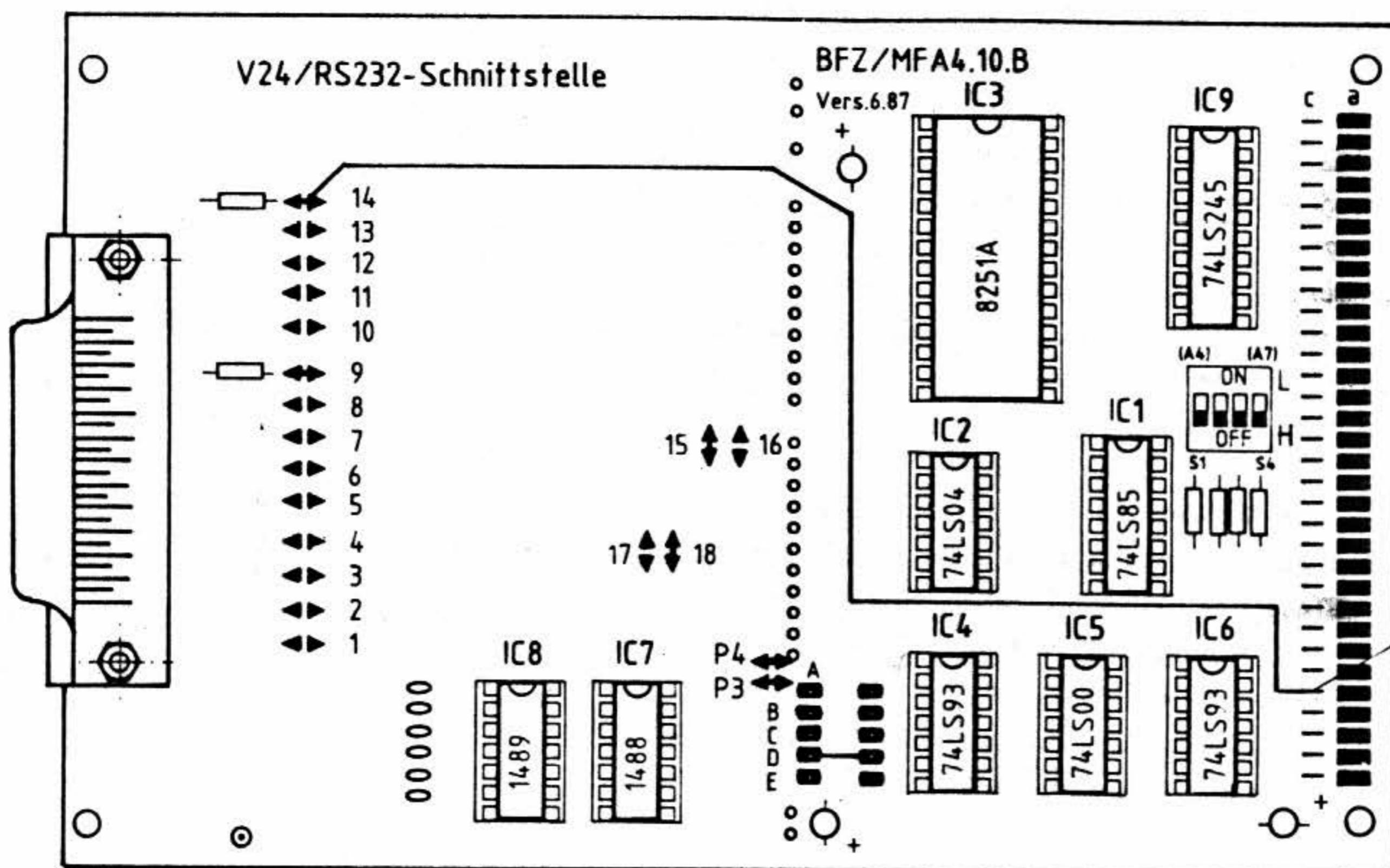


Bild 8: Bestückungsplan der V24/RS232-Schnittstelle

Für den Betrieb der Schnittstelle ist auf der Bestückungsseite der Schnittstelle die eingezeichnete Drahtbrücke nachzurüsten. Sie sollte mit Klebstoff fixiert werden.

## Einsatz unter CP/M

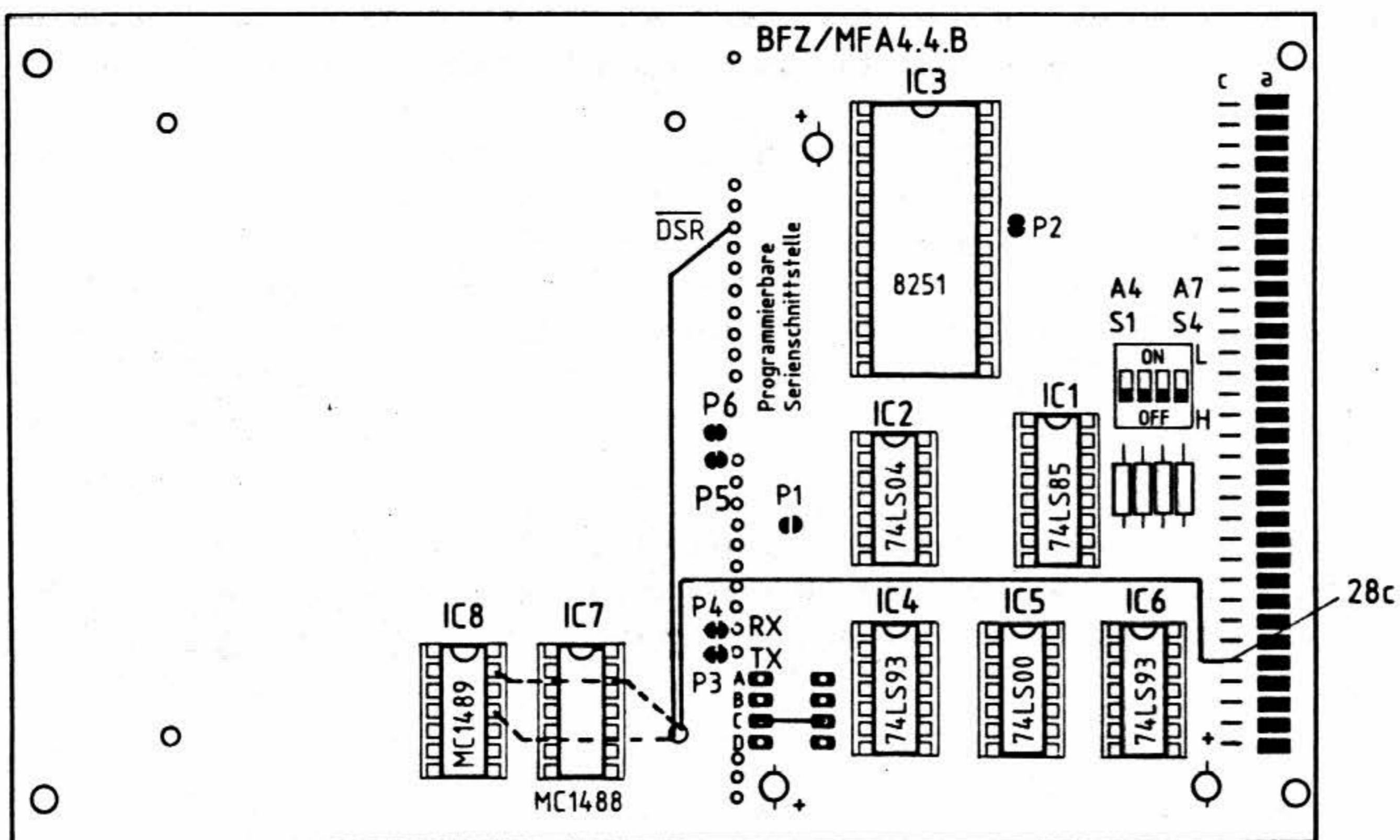


Bild 9: Bestückungsplan der Programmierbaren Serienschnittstelle

Für den Betrieb der Schnittstelle muß die Brücke zwischen dem Leiterplattenanschluß DSR und 0 V (GND) entfernt werden! Sie ist deshalb auf dem hier abgebildeten Bestückungsplan nicht eingezeichnet. Stattdessen sind auf der Bestückungsseite der Schnittstelle die zwei eingezeichneten Drahtbrücken nachzurüsten. Die Drahtbrücken sollten mit Klebstoff fixiert werden.

## Weitere Hinweise:

Der Datenaustausch zwischen der CPU und dem Video-Interface erfolgt über die zwischengeschaltete Serien-Schnittstelle. Diese ist notwendig, damit das CP/M-Betriebssystem eine Statusabfrage ("Ist ein Zeichen empfangen worden?" bzw. "Kann ein neues Zeichen gesendet werden?") durchführen kann.

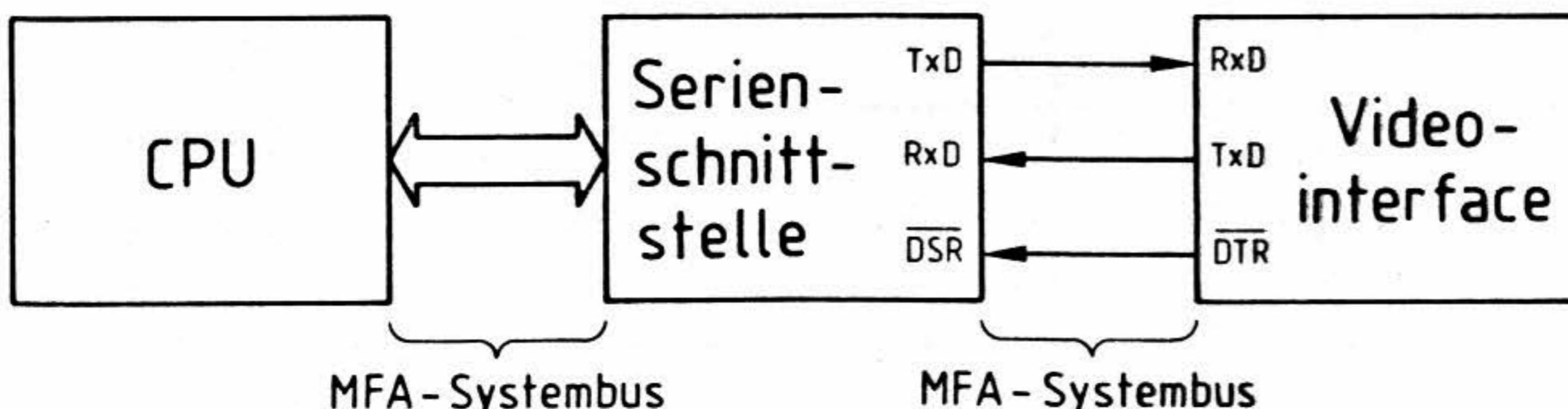


Bild 10: Datenaustausch zwischen CPU und Video-Interface (CP/M)

## Einsatz unter CP/M

Vor der Inbetriebnahme unter dem Betriebssystem CP/M sind die Schalter des Video-Interfaces entsprechend dem nachfolgenden Kapitel "Einstellung der Schalter des Video-Interfaces" einzustellen.

Die dort angegebene Schalterstellung gilt für eine Baudate von 1200 Bd, wie sie entsprechend FPÜ-Band 4 auf den oben erwähnten Serienschnittstellen eingestellt ist. Das hier beschriebene Video-Interface kann aber wahlweise auch mit höheren Baudaten betrieben werden. Mit der "Programmierbaren Serienschnittstelle" (BFZ/MFA 4.4) sind bis zu 4800 Bd, mit der "V24/RS232-Schnittstelle" (BFZ/MFA 4.10) bis zu 9600 Bd möglich. Die Baudatenbrücken der Serienschnittstellen und die DIL-Schalter des Video-Interfaces sind dann entsprechend zu ändern. Informationen über die Einstellung der Baudatenbrücken bzw. der DIL-Schalter können Sie den folgenden Unterlagen entnehmen:

- Für die "Programmierbare Serienschnittstelle" dem FPÜ-Band 2
- Für die "V24/RS232-Schnittstelle" den zur Baugruppe mitgelieferten Unterlagen
- Für das "Video-Interface" dem Anhang dieses Handbuches

Die Kapitel "Nutzung in Assembler Programmen" und "Nutzung unter MBASIC" enthalten Informationen für eigene Assembler- bzw. BASIC-Programme in denen die Steuerfunktionen des Video-Interfaces ausgenutzt werden sollen.

### 5.1 Einstellung der Schalter des Video-Interfaces

Auf der Interface-Baugruppe befinden sich drei Schalter-Reihen (siehe auch Bestückungsplan im Anhang) zur Grundeinstellung des Video-Interfaces beim Einschalten. Jede der mit S1, S2 und S3 bezeichneten Schalter-Reihen besteht aus acht Schaltern. S1.1 bezeichnet den Schalter Nummer 1 in der Schalter-Reihe S1. Die anderen Schalter sind entsprechend nummeriert. Für das Betriebsprogramm MAT 85 sind die Schalter folgendermaßen einzustellen:

#### - Schalter-Reihe S1:

Schalter	Stellung	eingestellte Funktion
S1.1	OFF	TVI 950-Mode
S1.2	OFF	blinkender Cursor
S1.3	ON	Block-Cursor
S1.4	OFF	dunkler Bildhintergrund
S1.5	OFF	reserviert, immer OFF !
S1.6	OFF	DTR-Handshake
S1.7	OFF	reserviert, immer OFF !
S1.8	OFF	Statuszeile aus

## Einsatz unter CP/M

## - Schalter-Reihe S2

Schalter	Stellung	eingestellte Funktion
S2.1	ON	Wortlänge 8 Bit
S2.2	OFF	Zeichensatz USA <sup>1)</sup>
S2.3	OFF	dto. <sup>1)</sup>
S2.4	OFF	dto. <sup>1)</sup>
S2.5	ON	Zeilenanzahl 24
S2.6	OFF	dto.
S2.7	ON	Spaltenanzahl 80
S2.8	OFF	dto.

- <sup>1)</sup> Dieser Zeichensatz wird für das Betriebssystem CP/M empfohlen, da hier häufig Zeichen benötigt werden, die im deutschen Zeichensatz nicht enthalten sind. (z.B.: "[" und "]" für Optionen beim PIP-Kommando). Sollten (z.B. für Textverarbeitung) andere Zeichensätze gewünscht werden, so kann der Zeichensatz über das Set-Up-Menü oder über eine Steuersequenz umgeschaltet werden, ohne die Einstellung der Schalter zu verändern. Viele CP/M-Programme (wie z.B. WORDSTAR) können eine solche Umschaltung beim Programmstart automatisch ausführen.

## - Schalter-Reihe S3

Schalter	Stellung	eingestellte Funktion
S3.1	ON	2 Stopbits
S3.2	OFF	kein Paritätsbit
S3.3	OFF	dto.
S3.4	OFF	dto.
S3.5	ON	1200 Baud <sup>2)</sup>
S3.6	OFF	dto. <sup>2)</sup>
S3.7	ON	dto. <sup>2)</sup>
S3.8	OFF	dto. <sup>2)</sup>

- <sup>2)</sup> Die hier angegebene Schalterstellung gilt für eine Baudrate von 1200 Bd, wie sie entsprechend FPÜ-Band 4 auf den Serienschnittstellen eingestellt ist. Falls die Serienschnittstellen mit einer anderen Baudrate betrieben werden, ist die Schalterstellung entsprechend zu ändern. Informationen über die Einstellung der DIL-Schalter können Sie dem Anhang dieses Handbuches entnehmen.

## Anmerkung:

Die Schalter S1.2 (Cursor-Art), S1.3 (Cursor-Form), S1.4 (Bildhintergrund), S1.8 (Statuszeile) und S2.2, S2.3, S2.4 (alle drei zur Einstellung des Zeichensatzes) können - den persönlichen Wünschen entsprechend - anders eingestellt werden. Eine andere Einstellung dieser Schalter beeinträchtigt die Funktion nicht. Angaben über die durch diese Schalter einstellbaren Optionen entnehmen Sie bitte dem Anhang.

---

Einsatz unter CP/M

## 5.2 Nutzung in Assembler-Programmen

### 5.2.1 Zeichenausgabe

Wenn der TVI 950-Mode unter dem Betriebssystem CP/M verwendet wird, können die BDOS-Funktionen des Betriebssystems zur Zeichenausgabe genutzt werden. Über diese Funktionen (Unterprogramme) können sowohl darstellbare Zeichen (wie Buchstaben und Ziffern) als auch Steuerzeichen ausgegeben werden.

Es gibt verschiedene BDOS-Funktionen zur Zeichenausgabe:

#### Funktion "CONSOLE OUTPUT", Ausgabe eines einzelnen Zeichens

Anwendung:

- C-Register mit  $02_{H}$  laden (Funktionscode)
- E-Register mit dem ASCII-Code des auszugebenden Zeichens laden
- BDOS mittels CALL 0005 aufrufen

#### Funktion "DIRECT CONSOLE I/O", Ausgabe eines einzelnen Zeichens

Anwendung:

- C-Register mit  $06_{H}$  laden (Funktionscode)
- E-Register mit dem ASCII-Code des auszugebenden Zeichens laden (der Code  $FF_{H}$  ist hier nicht zulässig)
- BDOS mittels CALL 0005 aufrufen

#### Funktion "PRINT STRING", Ausgabe einer Zeichenfolge

Anwendung:

- C-Register mit  $09_{H}$  laden (Funktionscode)
- DE-Registerpaar mit der Adresse laden, unter der das erste auszugebende Zeichen im Speicher steht. Die anderen auszugebenden Zeichen müssen in den darauffolgenden Speicherzeilen gespeichert sein. Die Zeichenfolge muß durch das "\$"-Zeichen (Code  $24_{H}$ ,  $36_{D}$ ) abgeschlossen sein. Das "\$"-Symbol selbst wird nicht ausgegeben.
- BDOS mittels CALL 0005 aufrufen

Beispiel:

```
ORG 0100H ; Anfangsadresse für CP/M-Programme
MVI E,42H ; Code 42H ("B") in das E-Register laden
MVI C,02H ; Funktionscode "CONSOLE OUTPUT"
CALL 0005H ; Zeichen ausgeben
MVI E,0AH ; Steuer-Code 0AH (Line Feed) laden
MVI C,06H ; Funktionscode "DIRECT CONSOLE I/O"
CALL 0005H ; Steuer-Code ausgeben
```

Einsatz unter CP/M

```
LXI D,TEXT ; Zeiger auf Zeichenfolge
MVI C,09H ; Funktionscode "PRINT STRING"
CALL 0005H ; Zeichenfolge ausgeben
JMP 0000H ; Programm-Ende
;-----
TEXT:    DB  'Dieser Text soll ausgegeben werden'
         DB  '$'      ; Endemarke (wird nicht ausgegeben)
```

Bitte beachten Sie, daß sämtliche Registerinhalte durch das BDOS verändert werden können. Gegebenenfalls sind die Registerinhalte vor dem BDOS-Aufruf (CALL 0005) mit Hilfe von PUSH-Befehlen im Stack zu speichern und nach der Ausführung der BDOS-Funktion durch POP-Anweisungen zu restaurieren.

Genauere Erläuterungen zu den BDOS-Funktionen entnehmen Sie bitte der CP/M-Literatur.

### 5.2.2 Zeicheneingabe

Alle Zeichen - gleichgültig ob sie über Tastatur eingegeben werden oder die "Antwort" auf eine DLE- bzw. Escape-Sequenz sind - können unter Verwendung von BDOS-Funktionen des Betriebssystems CP/M eingelesen werden.

Hierzu können mehrere BDOS Funktionen benutzt werden:

#### Funktion "CONSOLE INPUT", Einlesen eines Zeichens mit Echo

Anwendung:

- C-Register mit  $01_H$  laden (Funktionscode)
- BDOS mittels CALL 0005 aufrufen

Diese Funktion wartet, bis ein Zeichen eingegeben wird. Das vom Video-Interface zum Mikrocomputer gesendete Zeichen wird von der BDOS-Funktion sofort zum Interface zurück gesendet und auf dem Bildschirm angezeigt (Echo). Nach der Ausführung der BDOS-Funktion befindet sich das eingelesene Zeichen im Akkumulator der CPU. Diese BDOS-Funktion reagiert in der CP/M-üblichen Weise auf die Eingabe bestimmter Steuerzeichen:

Control S: Consolausgabe ein/aus

Control P: Drucker ein/aus

*Diese Funktion sollte nicht dazu verwendet werden. "Antworten" auf DLE- oder ESC-Sequenzen entgegenzunehmen, da alle vom Mikrocomputer empfangenen Zeichen auf dem Bildschirm angezeigt werden.*

## Einsatz unter CP/M

### Funktion "DIRECT CONSOLE I/O", Einlesen eines Zeichens ohne Echo

Anwendung:

- C-Register mit  $06_H$  laden (Funktionscode)
- E-Register mit  $FF_H$  laden (Input)
- BDOS mittels CALL 0005 aufrufen

Diese BDOS-Funktion ist eine kombinierte Ein-/Ausgabe-Funktion. Um die Eingabe zu wählen, muß das E-Register vor dem Funktionsaufruf mit dem Wert  $FF_H$  geladen werden. Diese Funktion führt - im Gegensatz zur oben beschriebenen Funktion "CONSOLE INPUT" - kein Echo aus. Das heißt: Das eingegebene Zeichen wird nicht auf dem Bildschirm angezeigt. Darüber hinaus wartet diese Funktion nicht auf eine Tastatur-Betätigung. Wurde keine Taste betätigt, so weist der Akkumulator nach der Rückkehr in das Anwenderprogramm den Inhalt  $00_H$  auf. Aus diesem Grund kann der Code  $00_H$  mit dieser Funktion nicht eingelesen werden. Enthält das Empfangsregister aufgrund einer Tastenbetätigung ein Zeichen, so befindet sich der entsprechende ASCII-Code im Akkumulator.

*Diese BDOS-Funktion sollte verwendet werden, wenn die "Antwort" auf eine DLE- oder Escape-Sequenz gelesen werden muß.*

### Funktion "READ CONSOLE BUFFER", Einlesen einer Zeile in einen Puffer

Anwendung:

- C-Register mit  $0A_H$  laden (Funktionscode)
- DE-Registerpaar mit Adresse des Eingabe-Puffers laden  
(Pufferlänge = Anzahl der zu puffernden Zeichen plus 2)
- Als erstes Byte in den Puffer die Anzahl der pufferbaren Zeichen eintragen (Pufferlänge-2).
- BDOS mittels CALL 0005 aufrufen

Diese BDOS-Funktion liest eine komplette Eingabezeile in den Puffer ein. Bei der Eingabe können alle üblichen CP/M-Editier-Funktionen genutzt werden (Backspace, Delete, ^C, ^E, ^H, ^J, ^M, ^R, ^U, ^X) Alle eingegebenen Zeichen werden auf dem Bildschirm angezeigt. Durch die Eingabe von "Line Feed" oder "Carriage Return" wird die Eingabe beendet. Das zweite Byte des Puffers enthält dann die Anzahl der tatsächlich eingegebenen Zeichen. Ab dem dritten Byte enthält der Puffer die eingegebenen Zeichen.

*Diese Funktion sollte nicht dazu verwendet werden, "Antworten" auf DLE- oder ESC-Sequenzen entgegenzunehmen, da alle vom Mikrocomputer empfangenen Zeichen auf dem Bildschirm angezeigt werden.*

Einsatz unter CP/M

## Beispiel:

Das folgende Programm zeigt im ersten Teil, wie Zeichen von der Tastatur gelesen werden können.  
Im zweiten Teil wird eine Abfrage der Software-Version gezeigt.

```
ORG 0100H ; Anfangsadresse fuer CP/M-Programme
LOOP: MVI C,01H ; Funktionscode "CONSOLE INPUT"
      CALL 0005H ; Lese Zeichen von der Tastatur in den Akku
                    ; und stelle es auf dem Bildschirm dar
      CPI 0DH ; Steuer-Zeichen "Carriage Return" eingel.?
      JNZ 'LOOP ; Wenn Nein --> LOOP
-----
; Hier weiter, wenn CR-Taste betätigt wurde
;

TEIL2: CALL NEWLIN ; Gebe CR/LF aus (neue Zeile)
      LXI D,VABFR ; Zeiger auf Steurseq. zur Vers.-Abfrage
      MVI C,09H ; Funktionscode "PRINT STRING"
      CALL 0005H ; Steuersequenz ausgeben
      MVI B,05H ; Laenge der Antwort (5): "V" x "/" y 0DH
      LXI H,BUF ; Zeiger auf Eingabe-Puffer
      CALL READ ; Lese Antwort
      MVI M,'$' ; Schließe Antwort mit '$' ab
      LXI D,BUF ; Zeiger auf Eingabe-Puffer
      MVI C,09H ; Funktionscode "PRINT STRING"
      CALL 0005H ; Zeige "Antwort" an
      JMP 0000H ; Programm-Ende
-----
; Unterprogramm, liest Zeichen ("Antwort") in Puffer ein
; Bei Aufruf:   HL = Puffer-Adresse
;                 B = Anzahl der zu lesenden Zeichen
; Nach UP-Ende: HL = Adr. des letzten gepufferten Zeichens + 1
;                 B = 0
;
READ: PUSH H
      PUSH B
WARTE: MVI C,06H ; Funktionscode "DIRECT CONSOLE I/O"
        MVI E,0FFH ; Waehle "Input"
        CALL 0005H ; Lese Eingabe
        ORA A ; Wurde ein Zeichen eingegeben ?
        JZ WARTE ; Nein --> WARTE
        POP B
```

## Einsatz unter CP/M

```
POP H
MOV M,A      ; Schreibe Zeichen in Puffer
INX H        ; Stelle Zeiger weiter
DCR B        ; Alle Zeichen gelesen ?
JNZ READ     ; Nein --> lese naechstes Zeichen
RET          ; Ja --> Return
;-----
; Unterprogramm, gibt CR/LF aus
;
NEWLIN: LXI D,CRLF ; Zeiger auf Steuerzeichen
         MVI C,09H   ; Funktionscode "PRINT STRING"
         CALL 0005H   ; Gebe Steuerzeichen aus
         RET
;-----
; Steuerzeichen, Steuer-Sequenzen
;
CRLF:    DB 0DH,0AH   ; Steuerzeichen CR/LF
         DB '$'       ; Ende-Marke
VABFR:   DB 10H,10H,'V' ; DLE-Sequenz
         DB '$'       ; Ende-Marke
;-----
; Puffer, 6 Byte
;
BUF:     DB 0,0,0,0,0,0 ; Puffer
```

Bitte beachten Sie, daß sämtliche Registerinhalte durch das BDOS verändert werden können. Gegebenenfalls sind die Registerinhalte vor dem BDOS-Aufruf (CALL 0005) mit Hilfe von PUSH-Befehlen im Stack zu speichern und nach der Ausführung der BDOS-Funktion durch POP-Anweisungen zu restaurieren.

Genauere Erläuterungen zu den BDOS-Funktionen entnehmen Sie bitte der CP/M-Literatur.

### 5.2.3 Ein Programmbeispiel in Assembler-Sprache

Das nachstehende Programm fordert den Anwender auf, eine Ziffer einzugeben. Der Aufforderungstext wird in dem Unterprogramm "TXTAUS" über die BDOS-Funktion "PRINT STRING" ausgegeben. Die Tastatur wird über die BDOS-Funktion "DIRECT CONSOLE I/O" abgefragt, da kein Echo erwünscht ist. Das Programm überprüft, ob tatsächlich eine Ziffer (ASCII-Code  $30_H$  bis  $39_H$ ) eingegeben wurde. Ist dies der Fall, gibt das Programm die entsprechende Ziffer als Wort auf dem Bildschirm aus. Wurde keine Ziffer eingegeben, erscheint die Meldung "KEINE ZIFFER". Das hier beschriebene Programm wertet die Betätigung der "CR"-Taste als Abbruchwunsch aus.

## Einsatz unter CP/M

```

        ORG  0100H      ;ANFANGSADRESSE FUER CP/M-PROGRAMME
START:  LXI  D,TEXT1    ;ZEIGER AUF 1. TEXT
        CALL TXTAUS    ;TEXT AUSGEBEN
WARTE: MVI  C,06H      ;FUNKTIONSCODE "DIRECT CONSOLE I/O"
        MVI  E,OFFH     ;WAEHLE "INPUT"
        CALL 0005H     ;LESE EINGABE (OHNE ECHO)
        ORA  A          ;ZEICHEN VON TASTATUR EINGELESEN ?
        JZ   WARTE     ;NEIN --> WARTE
        CPI  ODH        ;CR-TASTE BETAETIGT ?
        JZ   0000H     ;WENN JA --> ENDE
        CPI  30H        ;ASCII-CODE < 30H ?
        JC   NOZIFF    ;WENN JA --> NOZIFF
        CPI  3AH        ;ASCII-CODE >= 3AH ?
        JNC  NOZIFF    ;WENN JA --> NOZIFF
        SUI  2FH        ;ASCII-CODE AUF 1...10 (DEZ) REDUZIEREN
        MOV  B,A        ;ERMITTELTN WERT (ZAEHLER) NACH B
        LXI  D,ZIFF     ;ZEIGER AUF ZIFFERN-TEXTE
SUCHO:  LDAX D          ;ZEICHEN LESEN
        INX  D          ;ZEIGER AUF NAECHSTES ZEICHEN STELLEN
        CPI  '$'        ;'$' (TRENNZEICHEN) ?
        JNZ  SUCHO     ;WENN NEIN --> SUCHO
        DCR  B          ;ZAEHLER MINUS EINS
        JNZ  SUCHO     ;ZAEHLER NICHT NULL --> SUCHO
        CALL TXTAUS    ;TEXT AUSGEBEN
        JMP  START     ;AUF EIN NEUES
;-----
NOZIFF: LXI  D,NZTXT    ;ZEIGER AUF FEHLERQUELLE
        CALL TXTAUS    ;MELDUNG AUSGEBEN
        JMP  START     ;AUF EIN NEUES
;-----
TXTAUS: MVI  C,09H      ;FUNKTIONSCODE "PRINT STRING"
        CALL 0005H     ;GEBE ZEICHEN FOLGE AUS
        RET
;-----
TEXT1:  DB   ODH,0AH     ;WAGENRUECKLAUF, ZEILENVORSCHUB
        DB   'BITTE EINE ZIFFER EINGEBEN'
        DB   ODH,0AH     ;WAGENRUECKLAUF, ZEILENVORSCHUB
        DB   '$'         ;ENDEMARKE
NZTXT:  DB   07H         ;BELL
        DB   'KEINE ZIFFER'
        DB   ODH,0AH     ;WAGENRUECKLAUF, ZEILENVORSCHUB
        DB   '$'         ;ENDEMARKE

```

## Einsatz unter CP/M

---

ZIFF:	DB	'\$'	;MARKE
	DB	'NULL'	
	DB	'\$'	;MARKE
	DB	'EINS'	
	DB	'\$'	;MARKE
	DB	'ZWEI'	
	DB	'\$'	;MARKE
	DB	'DREI'	
	DB	'\$'	;MARKE
	DB	'VIER'	
	DB	'\$'	;MARKE
	DB	'FUENF'	
	DB	'\$'	;MARKE
	DB	'SECHS'	
	DB	'\$'	;MARKE
	DB	'SIEBEN'	
	DB	'\$'	;MARKE
	DB	'ACHT'	
	DB	'\$'	;MARKE
	DB	'NEUN'	
	DB	'\$'	;MARKE

---

## Einsatz unter CP/M

---

### 5.3 Nutzung unter MBASIC (BASIC 80)

MBASIC ist ein weit verbreiteter Basic-Interpreter für das Betriebssystem CP/M. In diesem Kapitel soll gezeigt werden, wie man die Eigenschaften des Video-Interfaces BFZ/MFA 8.4 in MBASIC-Programmen nutzen kann.

Das Betriebssystem CP/M benötigt nicht unbedingt ein Video-Interface zur Ausgabe von Zeichen, da es auch mit anderen Ausgabe-Geräten (z.B. einem Fernschreiber) zusammenarbeiten kann. Zu den möglichen Ausgabegeräten gehören auch solche, die am Zeilenende nicht automatisch Wagenrücklauf und Zeilenvorschub ausführen. Damit MBASIC auch mit solchen Geräten arbeitet, erzeugt es normalerweise nach jedem 80. ausgegebenen Zeichen die Steuerzeichenkombination Wagenrücklauf/Zeilenvorschub (CR/LF). Hierbei spielt es keine Rolle, ob die zuvor ausgegebenen Zeichen darstellbar waren (Buchstaben, Ziffern usw.) oder ob es sich um Steuerzeichen gehandelt hat. Das Video-Interface benötigt die von MBASIC nach jedem 80. Zeichen erzeugte CR/LF-Kombination nicht. Sie kann sich sogar störend auswirken, weil dadurch nur jede zweite Bildschirmzeile beschrieben wird. Fügt MBASIC die CR/LF-Kombination mitten in eine Steuer-Sequenz ein, so "versteht" das Interface die Sequenz nicht. Aus diesen Gründen muß das automatische Einfügen von CR/LF unterbunden werden. Dies kann durch den am Programmanfang stehenden Befehl

WIDTH 255

erreicht werden.

#### 5.3.1 Zeichenausgabe

Darstellbare Zeichen (wie Buchstaben und Ziffern) können unter MBASIC mit Hilfe der PRINT-Anweisung ausgegeben werden:

10 PRINT "Video-Interface"

Die Ausgabe von Steuerzeichen ist durch die PRINT- und CHR\$-Anweisungen möglich:

10 STEUERCODE=7 : PRINT CHR\$(STEUERCODE) ;  
20 PRINT CHR\$(7) ;

Die oben abgedruckten Zeilen sind gleichwertig, sie geben alle das Steuerzeichen BEL (ASCII-CODE 7<sub>D</sub>) aus. Die PRINT-Anweisungen sind jeweils mit einem ";" abgeschlossen, damit MBASIC nach dem auszugebenden Steuerzeichen BEL nicht zusätzlich die Steuerzeichen für Wagenrücklauf und Zeilenvorschub ausgibt.

Beachten Sie bitte die CHR\$-Anweisung! Ohne diese Anweisung würde die Ziffer "7" ausgegeben:

10 PRINT 7; : REM \*\*\* Diese Anweisung gibt die Ziffer "7" aus

## Einsatz unter CP/M

Sollen Steuerzeichen-Folgen (z.B. Escape-Sequenzen) ausgegeben werden, so kann hierzu eine einzige PRINT-Anweisung verwendet werden. Die einzelnen Steuerzeichen müssen - soweit sie nicht in einem *String* zusammengefaßt sind - durch ein ";" (**nicht durch ein "," !**) voneinander getrennt werden:

```
10 PRINT CHR$(27);CHR$(71);CHR$(52);
```

Das Semikolon zwischen den einzelnen Zeichen verhindert, daß MBASIC Leerzeichen in die zum Video-Interface übertragene Zeichenfolge einfügt.

Die oben angegebene Zeichenfolge schaltet die inverse Zeichendarstellung ein. Auch andere Möglichkeiten zur Ausgabe von Steuerzeichenfolgen sind erlaubt:

```
10 PRINT CHR$(27);"G4";
```

oder

```
10 ESC$=CHR$(&H1B)
20 PRINT ESC$;"G4";
```

oder

```
10 INVEIN$=CHR$(27)+"G4"
20 PRINT INVEIN$;
```

usw.

### 5.3.2 Zeicheneingabe

Um Tastatur-Eingaben in MBASIC entgegenzunehmen, können die Basic-Anweisungen INPUT, INPUT\$ und INKEY\$ verwendet werden. Eine genaue Beschreibung dieser Anweisungen entnehmen Sie bitte der MBASIC-Anleitung.

Beispiele:

```
10 INPUT "Eingabe", I1$
20 I2$=INPUT$(4) : REM *** LIEST 4 ZEICHEN
30 I3$=INKEY$ : IF I3$="" THEN 30
```

"Antworten" auf DLE- bzw. Escape-Sequenzen sollten ausschließlich mit der INPUT\$-Anweisung entgegengenommen werden. Dabei sollte - wie das nachfolgende Beispiel zeigt - die INPUT\$-Anweisung direkt auf die PRINT-Anweisung zur Ausgabe der DLE- bzw. Escape-Sequenz folgen. Bitte achten Sie darauf, daß in der Klammer der INPUT\$-Anweisung die richtige Anzahl der "Antwort"-Zeichen angegeben wird.

Einsatz unter CP/M

```

1 WIDTH 255 : REM *** UNTERDRUECKE VOM INTERPRETER-CR/LF
10 ESC$=CHR$(&H1B) : REM *** ESCAPE
20 WO$=ESC$+"?" : REM *** ESC-SEQUENZ ZUR CURSOR-POSITIONS-ABFR.
30 REM *** FUER DIESES BEISPIEL IST ES WICHTIG, DASS DIE NACH-
40 REM *** FOLGENDE PRINT-ANWEISUNG MIT EINEM ";" ABGESCHLOSSEN
50 REM *** IST.
60 PRINT "Bitte CONTROL-C betaetigen"; : REM *** Text ausgeben
70 PRINT WO$; : REM *** ABFRAGE DER CURSOR-POSITION (mit ";" !)
80 I$=INPUT$(3) : REM *** "ANTWORT" ENTGEGENNEHMEN (3 ZEICHEN)
90 REM *** DIE ANTWORT Besteht AUS 3 ZEICHEN:
100 REM *** REIHE+20H, SPALTE+20H, ODH
110 REIHE=ASC(LEFT$(I$,1))-&H20
120 SPALTE=ASC(MID$(I$,2,1))-&H20
130 REM *** NUN DIE CURSOR-POSITION ANZEIGEN
140 PRINT "Reihe=";REIHE; , Spalte=";SPALTE
150 REM *** NUN CURSOR AUF DIE ALTE POSITION, DIE ER VOR DER
160 REM *** ANZEIGE VON "REIHE" UND "SPALTE" HATTE
170 PRINT ESC$;"=";LEFT$(I$,2);
180 REM *** DIE ERSTEN BEIDEN ZEICHEN VON I$ GEBEN DIE POSITION AN
190 REM *** ZUM ABSCHLUSS EINE ENDLOS-SCHLEIFE, DAMIT DIE
200 REM *** CURSOR-POSITION NICHT DURCH DIE READY-MELDUNG DES
210 REM *** BASIC-INTERPRETERS VERAENDERT WIRD (PROGRAMM BITTE
220 REM *** MIT "CONTROL-C" ABBRECHEN)
230 GOTO 230

```

**5.3.3 Ein Programmbeispiel in MBASIC (BASIC 80)**

Das nachstehende Programm soll die einzelnen Aus- und Eingabemöglichkeiten unter MBASIC darstellen. Texte und Steuerzeichen werden mit Hilfe einer PRINT-Anweisung ausgegeben. Zeile 190 zeigt die Anwendung der INKEY\$-Funktion.

```

1 WIDTH 255 : REM *** UNTERDRUECKE INTERPRETER-CR/LF
10 PRINT CHR$(7); : REM *** CODE FUER "BEL" AUSGEBEN
20 PRINT "Dies ist ein Test" : REM *** TEXT AUSGEBEN
30 PRINT : REM EINE LEERZEILE ERZEUGEN
40 PRINT "Gross- und Kleinbuchstaben"
50 REM *** NUN DEN CURSOR ZWISCHEN DIE BEIDEN TEXTZEILEN STELLEN

```

## Einsatz unter CP/M

```
60 REM *** (DER CURSOR STEHT NACH DER LETZTEN PRINT-ANWEISUNG
70 REM *** UNTER DEM TEXT "GROSS - ....". DAHER MUSS DER CURSOR
80 REM *** UM ZWEI ZEILEN NACH OBEN GESTELLT WERDEN. DER CODE
90 REM *** FUER "CURSOR HOCH" IST 11 (DEZ))
100 PRINT CHR$(11);CHR$(11);
110 REM *** ACHTEN SIE BEIM EINGEBEN DIESES PROGRAMMS DARAUF,
120 REM *** DASS AM ENDE DER NACHFOLGENDEN ZEILE EIN ";" STEHT
130 PRINT "-----";
140 REM *** NUN CURSOR UNTER "Gross- und Kleinbuchstaben"
150 PRINT CHR$(10);CHR$(10);CHR$(13);: REM *** LF, LF, CR
160 INPUT "WIE ALT SIND SIE ",A
170 PRINT "SIE BEHAUPTEN,";A;"JAHRE ALT ZU SEIN"
180 PRINT "BITTE BETAETIGEN SIE EINE TASTE"
190 C$=INKEY$ : IF C$="" THEN 190 : REM *** TASTATURABFRAGE
200 PRINT "DIE VON IHNEN BETAETIGTE TASTE ERZEUGT DEN CODE";
210 PRINT ASC(C$);" (DEZ)" : REM *** CODE ANZEIGEN
220 PRINT "--- ENDE ---"
```

#### 5.4 Das Editor-Programm BFZED.COM

Der zum CP/M-Betriebssystem für den MFA-Mikrocomputer mitgelieferte Editor "BFZED.COM" ist für das Video-Interface BFZ/MFA 8.2 installiert. Dies bedeutet, daß zur Steuerung des Cursors der MAT 85-Mode des hier beschriebenen Interfaces benutzt werden muß. Dieser Mode kann vor dem Start des Editors über das Set-Up-Menü ein- und nach Verlassen des Editors wieder ausgeschaltet werden. Abgesehen davon, daß hierbei ständig zwischen den beiden Modi hin- und hergeschaltet werden muß, können die erweiterten Möglichkeiten des TVI 950-Modus so nicht genutzt werden.

Deshalb ist es sinnvoll, das Programm BFZED.COM an den TVI 950-Mode anzupassen. Bevor man diese Anpassung jedoch durchführt, sollte man sich eine Sicherheitskopie des Programms anfertigen.

#### **Arbeiten Sie NIE mit der Original-Diskette (außer zum Erstellen einer Sicherheitskopie)**

Bedenken Sie ebenfalls, daß die geänderte BFZED.COM-Version ausschließlich im TVI 950-Mode des hier beschriebenen Video-Interfaces BFZ/MFA 8.4 funktionsfähig ist. Sollten Sie BFZED.COM auch auf dem Video-Interface BFZ/MFA 8.2 betreiben wollen, so müssen Sie hierfür die ungeänderte Version des Programms benutzen.

Einsatz unter CP/M**5.4.1 Vorbereitungen zum Anpassen des Programms BFZED.COM**

- Kopieren Sie folgende Programme auf eine formatierte und mit dem CP/M-Betriebssystem versehene Diskette: STAT.COM, DDT.COM, BFZED.COM und BFZED.HLP.  
**Alle Arbeiten sollten nur mit dieser neu geschaffenen Arbeitsdiskette durchgeführt werden, damit nicht versehentlich die Original-Version des Editors verändert wird.**
- Setzen Sie die neue Arbeitsdiskette in Laufwerk A: ein und betätigen Sie die RESET-Taste des Mikrocomputers. Nach wenigen Sekunden muß die CP/M-Bereitschaftsmeldung "A>" erscheinen.
- Ermitteln Sie die Länge des Programms BFZED.COM mit Hilfe des Programms STAT.COM. Geben Sie hierzu die Anweisung

STAT BFZED.COM

ein und schließen Sie die Eingabe durch die Betätigung der CR-Taste ab.

Auf dem Bildschirm wird daraufhin beispielsweise

RECS	BYTES	EXT	ACC
64	8K	1	R/W A:BFZED.COM
BYTES REMAINING ON A: 288K			

angezeigt. Hiervon ist nur der Wert unter "RECS" wichtig. Er gibt die Länge des Programms BFZED.COM in Records zu je 128 Bytes an. Falls der angezeigte Wert ungerade ist, muß er um 1 erhöht werden. Dividieren Sie den (ggf. um 1 erhöhten) Wert durch 2 und notieren Sie das Divisions-Ergebnis.

Beispiele:

Angezeigter Wert	plus 1 (da ungerade)?	Divisions-Ergebnis (Notieren !)
63	JA --> 64	32
64	NEIN	32

**5.4.2 Anpassen des Programms BFZED.COM**

- Geben Sie nach den oben angegebenen Vorbereitungen nun die folgende Kommandozeile ein:

DDT BFZED.COM

Schließen Sie die Eingabe mit der CR-Taste ab. Die Eingabe bewirkt, daß zuerst das Monitorprogramm DDT.COM geladen wird. Dieses lädt dann das Editor-Programm BFZED.COM in den RAM-Speicher. Dort kann es verändert werden.

Nachstehend ist der "Dialog" zwischen Benutzer und DDT.COM abgedruckt. Ausgaben des Programms DDT.COM sind in der Schrift Courier gesetzt. Eingaben des Benutzers sind in der

## Einsatz unter CP/M

unterstrichenen Schrift ***Courier*** gesetzt. Alle Benutzereingaben sind durch die Betätigung der CR-Taste abzuschließen. Das Zeichen "♂" im folgenden Bildschirm-Dialog symbolisiert DDT-Ausgaben, die - abhängig von der jeweiligen BFZED.COM-Version - unterschiedlich sein können. Kommentare sind in der Schrift *Times Roman Italic* gesetzt.

DDT VERS ♂.♂ *Meldung des Programms DDT.COM*

NEXT PC

♂♂♂♂ ♂♂♂♂

-S0106

*Benutzereingabe mit CR-Taste abschließen !*

0106 ♂♂ 4D *Anzahl der Zeichen pro Zeile (hex.)*

0107 ♂♂ 17 *Anzahl der Zeilen pro Bildschirmseite (hex.)*

0108 ♂♂ 0 *Keine Zeitverzögerung nach dem Löschen d. Bildschirms*

0109 ♂♂ + *Punkt "." eingeben, anschließend auch hier CR betätigen*

-S012F

012F ♂♂ 1 *Ein Steuerbyte für Cursor nach rechts*

0130 ♂♂ 0C *Steuerbyte: ^L*

0131 ♂♂ + *Punkt "." eingeben, anschließend auch hier CR betätigen*

-S013E

013E ♂♂ 1 *Ein Steuerbyte für Bildschirm löschen, Home Cursor*

013F ♂♂ 1A *Steuerbyte: ^Z*

0140 ♂♂ + *Punkt "." eingeben, anschließend auch hier CR betätigen*

-S0143

0143 ♂♂ 3 *Drei Steuerbyte für Carriage Return mit Löschen des Zeilenrestes*

0144 ♂♂ 1B *1. Steuerbyte: Escape*

*2. Steuerbyte: ASCII-Code für "T"*

0145 ♂♂ 54 *3. Steuerbyte: Wagenrücklauf*

0146 ♂♂ 0D *Punkt "." eingeben, anschließend auch hier CR betätigen*

0147 ♂♂ +

*Punkt "." eingeben, anschließend auch hier CR betätigen*

-S0148

0148 ♂♂ 0D *Steuerzeichen für Carriage Return ohne Löschen des Zeilenrestes*

0149 ♂♂ + *Punkt "." eingeben, anschließend auch hier CR betätigen*

-

- Nachdem alle Änderungen durchgeführt wurden, kann das Programm DDT.COM durch die Tastenkombination CONTROL-C beendet werden.
- Nun ist es wichtig, die noch im RAM befindliche neue Version des Programms BFZED.COM auf der Diskette zu speichern. Erinnern Sie sich noch an den Wert, den Sie sich notieren sollten? Diesen Wert brauchen wir jetzt! Geben Sie die Kommando-Zeile

SAVE ♂♂ BFZED950.COM

ein ("♂♂" symbolisiert den notierten Wert). Das Betriebssystem CP/M "weiß" nun, wie groß der RAM-Bereich ist, der auf der Diskette gespeichert werden muß. Die neue Version des Editor-Programms wird unter dem Namen BFZED950.COM gespeichert.

- Anschließend kann die neue Version des Editors aufgerufen und überprüft werden.

Einsatz unter CP/M

- Die hier durchgeföhrten Änderungen passen den Editor lediglich an das neue Bildformat (80 Zeichen pro Zeile, 24 Zeilen) an. Soll der Editor z.B. beim Programmaufruf automatisch vom USA-Zeichensatz zum deutschen Zeichensatz umschalten, so ist auch dies möglich. Die Steuerzeichen für die zusätzlichen Anpassungen entnehmen Sie bitte diesem Handbuch. Der Änderungsvorgang selbst ist im FPÜ-Band 4 ("Beschreibung des Programms BFZED.COM") erläutert.

## Die Statuszeile

### 6 Die Statuszeile

Das hier beschriebene Interface kann zusätzlich zu den Zeichen-Zeilen am unteren Rand des Schriftfeldes eine Statuszeile anzeigen. Diese kann mit dem Schalter S1.8 oder mit Hilfe des Set-Up-Menüs ein- bzw. ausgeschaltet werden. Die Statuszeile ist unabhängig vom eingeschalteten Mode (MAT 85 bzw. TVI 950). Sie steht dem Anwender nicht zur Zeichenausgabe zur Verfügung, das heißt, er kann in diese zusätzliche Zeile keinen Text schreiben. In der Statuszeile wird mit halber Helligkeit und dunkler Schrift auf hellem Grund eine Zustandsinformation des Video-Interfaces angezeigt.

Die Statuszeile besteht aus acht Spalten und hat folgenden Aufbau:

Control-Mode	Parität	Datenbits	Stopbits	Baudrate	Zeichensatz	Emulations-Mode	Rrr Css
CTL-MODE	P:OFF	D:8	S:2	1200 BAUD	USA	TVI950	R02 C17

Die Überschriften der einzelnen Spalten werden auf dem Bildschirm nicht angezeigt. Die hier dargestellte Anzeige ist ein Beispiel. Die tatsächlich angezeigten Werte sind abhängig von der Einstellung des Video-Interfaces bzw. von der Cursor-Position.

#### Control-Mode:

Wenn der Control-Mode aktiviert wurde, zeigt das Interface in dieser Spalte die Meldung "CTL-MODE" an. Normalerweise ist dieses Feld leer.

#### Parität:

Hinter den Zeichen "P:" wird der aktuelle Paritäts-Mode angezeigt. Möglich sind die Anzeigen "OFF" (keine Parität), "ODD" (odd Parity, ungerade Parität), "EVN" (even Parity, gerade Parität), "MRK" (Mark, immer H-Pegel), "SPC" (Space-Parity, immer L-Pegel). Der Paritäts-Mode kann über die DIL-Schalter der Baugruppe eingestellt werden.

#### Datenbits:

In dieser Spalte wird hinter den Zeichen "D:" die Anzahl der Datenbits pro übertragenem Zeichen angezeigt. Möglich sind "7" bzw. "8" Datenbits. Die Anzahl der Datenbits kann über die DIL-Schalter der Baugruppe eingestellt werden.

#### Stopbits:

Die eingestellte Anzahl der Stopbits ("1" oder "2") wird hinter den Zeichen "S:" angezeigt. Die Anzahl der Stopbits kann über die DIL-Schalter der Baugruppe eingestellt werden.

#### Baudrate:

Die eingestellte Baudrate wird in dieser Spalte angezeigt. Folgende Baud-Raten sind möglich: 110 Bd; 134,5 Bd; 150 Bd; 300 Bd; 600 Bd; 1200 Bd; 1800 Bd; 2400 Bd; 3600 Bd; 4800 Bd; 7200 Bd; 9600 Bd; 19200 Bd.

---

Die Statuszeile

---

**Zeichensatz:**

Der aktuelle länderspezifische Zeichensatz wird hier durch eine entsprechende Abkürzung angezeigt:

<u>USA</u>	Vereinigte Staaten von Amerika
<u>FRANCE</u>	Frankreich
<u>GERmany</u>	Deutschland
<u>ENGLand</u>	England
<u>DENmark</u>	Dänemark
<u>SWEden</u>	Schweden
<u>ITALy</u>	Italien
<u>SPAin</u>	Spanien

**Emulations-Mode:**

In dieser Spalte wird der aktuelle Emulations-Mode ("MAT85" oder "TVI950") angezeigt.

**Rrr Css (Cursor-Position):**

Im rechten Teil der Status-Zeile wird die aktuelle Cursor-Position angezeigt. Hinter "R" (für "ROW", Reihe) wird die Nummer der Reihe, in der sich der Cursor momentan befindet, angezeigt. Hinter "C" (für "COLUMN", Spalte) erfolgt die Anzeige der Spalten-Nummer. Die für diese Anzeige gültige Reihen- und Spaltenzählung beginnt bei 1.

**Hinweis:**

Wie beim original TVI 950-Terminal, beginnt die Spalten- und Reihenzählung zur Abfrage der Cursor-Position (ESC ?) und zur Positionierung des Cursors (ESC =) bei 0, die Zählung für die Cursor-Positionsanzeige in der Status-Zeile aber bei 1.

## Das Set-Up-Menü

### 7 Das Set-Up-Menü

Viele Einstellungen, die über DIL-Schalter bzw. Escape-Sequenzen möglich sind, lassen sich auch über die Tastatur verändern. Hierzu muß das *Set-Up-Menü* (Einstell-Menü) durch die Eingabe von **CONTROL-DEL** aufgerufen werden (CONTROL-Taste gedrückt halten, zusätzlich die im rechten Teil der Tastatur befindliche DEL-Taste betätigen (nicht die DEL-Taste links oben auf der Tastatur), anschließend beide Tasten loslassen).

Nach dem Aufruf des Set-Up-Menüs erscheint folgende Anzeige auf dem Bildschirm:

<u>SETUP Video-Interface BFZ/MFA 8.4 Vx.x</u>			
Auswahl mit "Auf"/"Ab", ändern mit Leertaste, Ende mit CONTROL <u>DEL</u>			
<u>TERMINAL</u>		<u>VIDEO</u>	
Emulation	TV1950	Hintergrund	DUNKEL
Spalten	80	Attr. Bit 7	AUS
Zeilen	24	<u>CURSOR</u>	
Zeichensatz	USA	Form	BLOCK
Statuszeile	AUS	Art	BLINK
Control Mode	AUS		

Die oben dargestellte Bildschirm-Anzeige zeigt eine mögliche Einstellung des Video-Interfaces. Je nach der Einstellung des von Ihnen verwendeten Interfaces kann die Anzeige an einigen Stellen davon abweichen.

Die Anzeige soll nun im einzelnen erläutert werden:

#### SETUP Video-Interface BFZ/MFA 8.4 Vx.x

Die erste Zeile ist die Meldung des Set-Up-Menüs. In ihr wird auch die Version des auf dem Interface verwendeten EPROMs angezeigt (hier mit "vx.x" angedeutet).

Die zweite Zeile gibt einige Bedienungshinweise:

Auswahl mit "Auf"/"Ab", ändern mit Leertaste, Ende mit CONTROL DEL

---

Das Set-Up-Menü

---

Der restliche Teil des Set-Up-Menüs ist in zwei Spalten aufgeteilt. In der ersten Zeile der linken Spalte befindet sich der Text

Emulation	TVI950
-----------	--------

Der Textteil hinter dem Wort "Emulation" (entweder - wie hier - TVI950 oder MAT85) ist invers dargestellt. Dieser inverse Block kann über die Pfeiltasten "Auf" und "Ab" der Tastatur in die anderen Zeilen der Spalte bewegt werden (versuchen Sie es einmal!). Befindet sich der inverse Block in der untersten Zeile einer Spalte, springt er bei Betätigung der "Ab"-Taste in die oberste Zeile der Nachbar-Spalte. Betätigt man nun die "Auf"-Taste, springt er zurück in die alte Position. Durch das Positionieren des inversen Blocks kann man auswählen, welcher Parameter (Emulation, Spalten, Zeilen ...) geändert werden soll.

Eine Änderung der einzelnen Parameter wird durch Betätigung der Leertaste erreicht. Bei jeder Betätigung wird eine neue Einstellung innerhalb des inversen Blocks angezeigt. Die Leertaste muß so oft betätigt werden, bis die gewünschte Einstellung erscheint. Sind auf diese Weise alle möglichen Einstellungen eines Parameters der Reihe nach zur Anzeige gebracht worden, erscheint wieder die erste Einstellung. Die eingestellten Parameter werden erst beim Verlassen des Set-Up-Menüs aktiv.

Wenn alle Parameter entsprechend den persönlichen Wünschen eingestellt sind, kann das Set-Up-Menü durch erneute Betätigung von CONTROL-DEL verlassen werden. Dabei wird der alte Bildschirminhalt - soweit nach der Parameteränderung möglich - restauriert, und die Zeichendarstellung den aktuellen Parametern angepaßt. Eine Änderung des länderspezifischen Zeichensatzes wirkt sich erst bei der Übertragung weiterer Zeichen aus.

Folgende Einstellungen sind möglich:

Emulation	:	TVI950 / MAT85
Spalten	:	72 / 80 / 88 / 96
Zeilen	:	22 / 24 / 26 / 28
Zeichensatz	:	USA / FRA / GER / ENG / DEN / SWE / ITA / SPA
Statuszeile	:	AUS / EIN
Control Mode	:	AUS / EIN
Hintergrund	:	HELL / DUNKEL
Attribut Bit 7	:	AUS / REVERS / BLINK / UNDERL
Cursor-Form	:	STRICH / BLOCK
Cursor-Art	:	BLINK / RUHEND / AUS

## Der Control-Mode

### 8 Der Control-Mode

Im Control-Mode werden alle **Steuerzeichen**, die das Video-Interface empfängt, nicht ausgeführt, sondern als inverse Zeichen mit halber Helligkeit auf dem Bildschirm angezeigt. Darstellbare Zeichen werden unverändert angezeigt. Mit Hilfe dieses Modes kann der Anwender feststellen, welche Steuerzeichen vom Mikrocomputer zum Video-Interface gesendet werden. So ist eine einfache Überprüfung der Anwender-Software möglich.

Die folgende Tabelle zeigt einige häufig benutzte Steuerzeichen und ihre Darstellung (eine vollständige Tabelle finden Sie im Anhang):

Code	Mnemonik	angez. Zeichen <sup>1)</sup>
07 <sub>H</sub>	BEL	G
08 <sub>H</sub>	BS	H
09 <sub>H</sub>	HT	I
0A <sub>H</sub>	LF	J
0B <sub>H</sub>	VT	K
0C <sub>H</sub>	FF	L
0D <sub>H</sub>	CR	M
10H	DLE	P
16 <sub>H</sub>	SYN	V
1A <sub>H</sub>	SUB	Z
1B <sub>H</sub>	ESC	[
1E <sub>H</sub>	RS	^

<sup>1)</sup> Die angezeigten Zeichen sind abhängig vom eingestellten länderspezifischen Zeichensatz. Für diese Tabelle wurde der USA-Zeichensatz (empfohlene Grundeinstellung und Tastaturschriftung) verwendet.

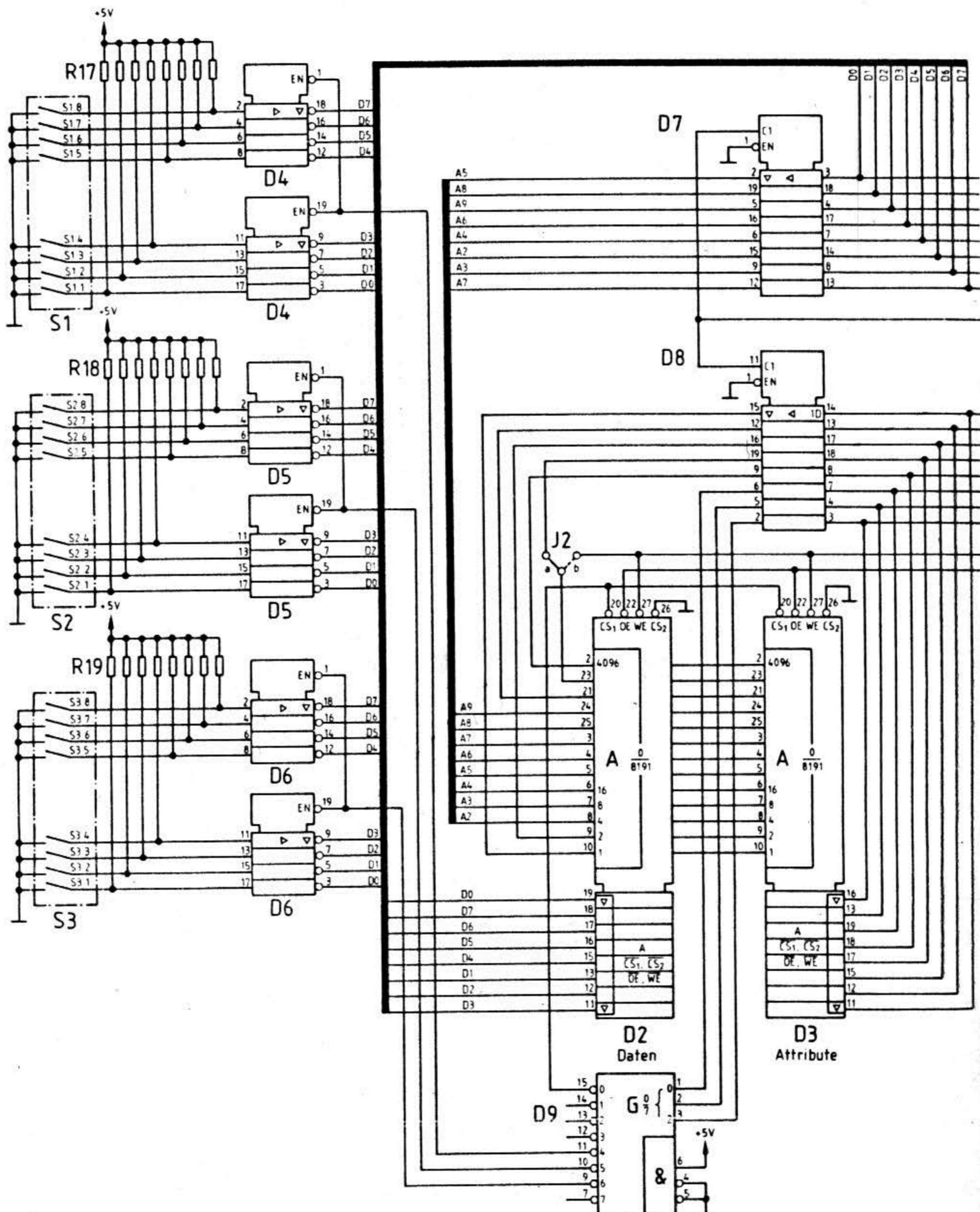
Der Control-Mode kann auf zwei verschiedene Arten aktiviert werden:

- durch Einschalten von "Control Mode EIN" im Set-Up-Menü
- durch Senden der Escape-Sequenz "ESC U" im TVI 950-Mode

#### Hinweis:

Da im Control-Mode alle Steuerzeichen nur noch angezeigt aber nicht ausgeführt werden, kann der Control-Mode nur über das Set-Up-Menü (Control Mode AUS) verlassen werden. Siehe auch: "Das Set-Up-Menü".

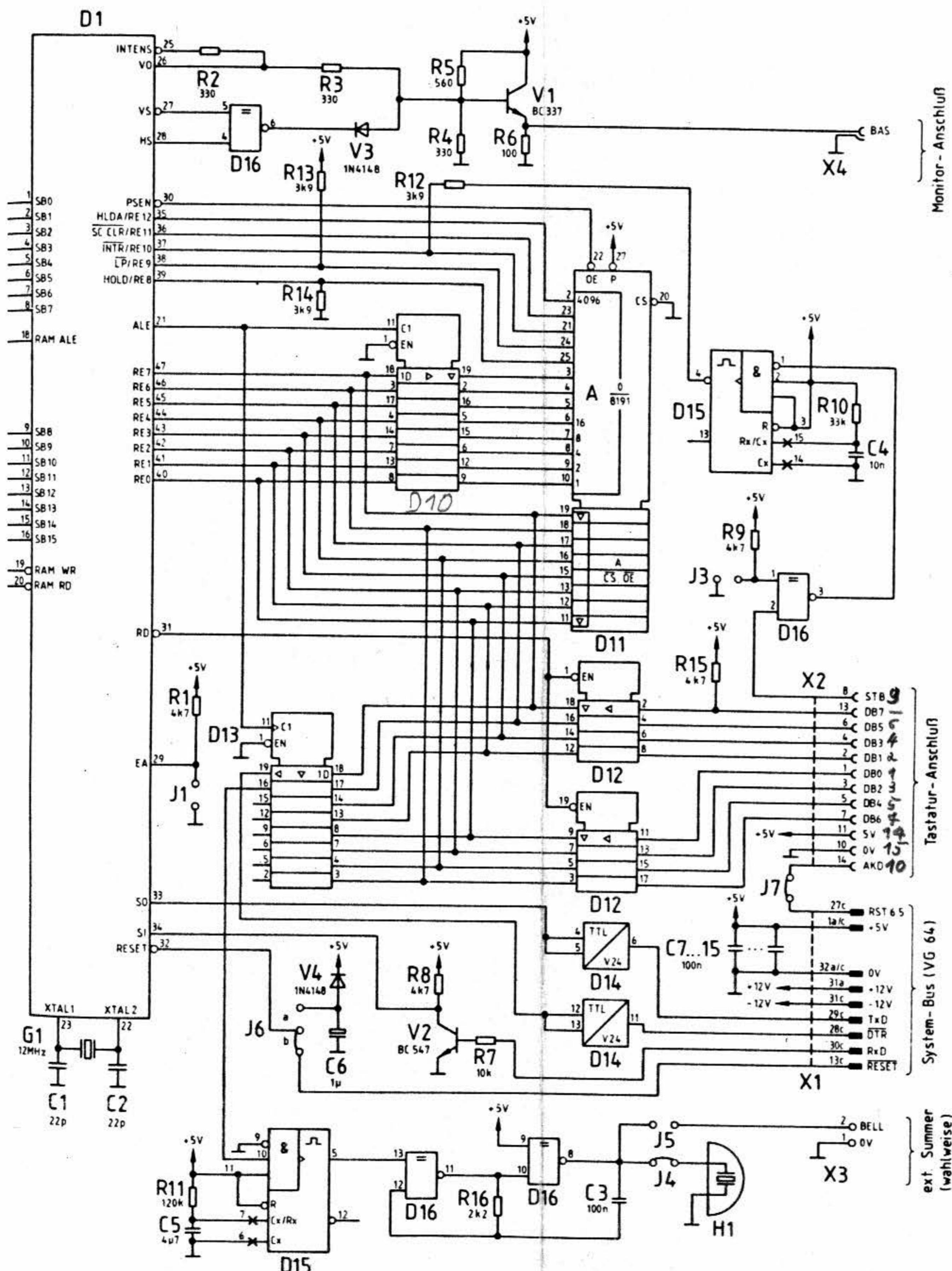
## Anhang 1: Stromlaufplan



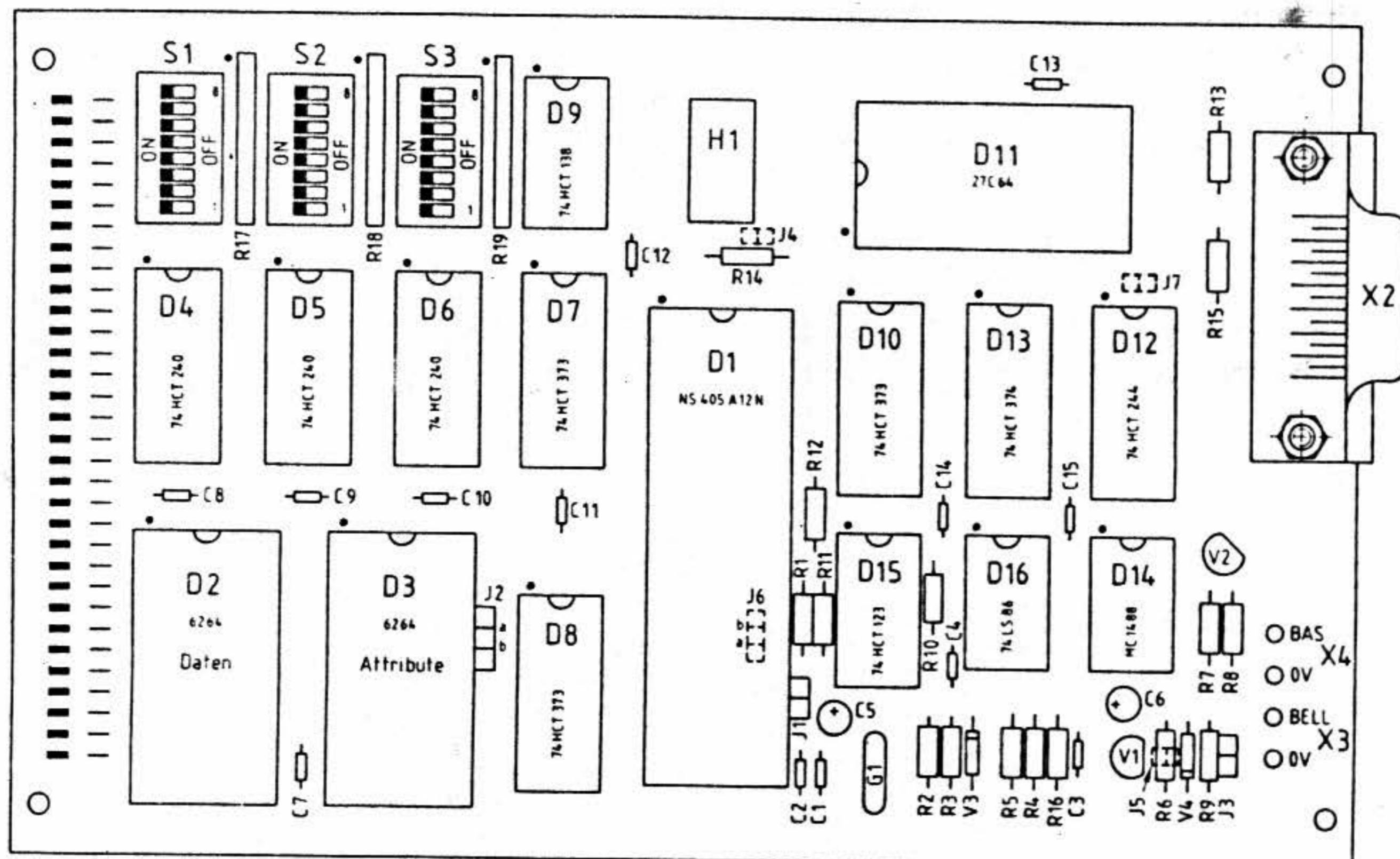
D1	D2, D3	D4, D6	D7, D8, D10	D9	D11	D12	D13	D14	D15	D16
NS405A12N	6264	74HCT240	74HCT373	74HCT138	27C64	74HCT244	74HCT374	MC1488	74HCT123	74LS86
0V	24	14	10	10	8	14	10	10	7	8
+5V	48	28	20	20	16	1,28	20	20	-	16
+12V	-	-	-	-	-	-	-	-	74	-
-12V	-	-	-	-	-	-	-	-	1	-

R17 R19 SIL - Widerstandnetzwerk 4k7

## Anhang 1: Stromlaufplan



## Anhang 2: Bestückungsplan



## Anhang 3: Bauteilliste

elektr. Bez.	Art	Bemerkung
H1	Piezokeramischer Signalgeber B/C 11-S	
C1, C2	Keramik-Kondensator 22 pF, 50 V	
C3, C7...C15	Keramik-Kondensator 0,1 uF, 50 V	
C4	Keramik-Kondensator 10 nF, 50 V	
C5	Tantal-Elko 4,7uF; 35V	Tropfenform
C6	Tantal-Elko 1uF; 35V	Tropfenform
D1	Terminal Management Prozessor NS 405 A12N	
D2, D3	RAM-Baustein 8K x 8 Bit, statisch max. 300ns, pinkompatibel mit 6264	
D4...D6	8 inv. Bus-Leitungstreiber 74 HCT 240	Tri-State
D7, D8, D10	8-Bit-D-Latch 74 HCT 373	Tri-State
D9	3-Bit-Binärdekoder/Demultiplexer 74 HCT 138	
D11	EPROM-Baustein 8K x Bit, CMOS max. 300ns, pinkompatibel mit 2764	programmiert mit der Software des Video-Interfaces
D12	8 Bus-Leitungstreiber 74 HCT 244	Tri-State
D13	8-Bit-D-Register 74 HCT 374	Tri-State
D14	4 Leitungstreiber für V24-Schnittstellen MC 1488 oder 75188	
D15	2 retriggerbare Monoflops 74 HCT 123	
D16	4 Exklusiv-Oder-Gatter je 2 Eing. 74 LS 86	kein HCT verwenden!
G1	Quarz 12 MHz, HC18U stehend	
R1, R8, R9, R15	Widerstand 4,7 kOhm	Alle Widerstände 0,25W; ±5%
R2, R3, R4	Widerstand 330 Ohm	
R5	Widerstand 560 Ohm	
R6	Widerstand 100 Ohm	
R7	Widerstand 10 kOhm	
R10	Widerstand 33 kOhm	
R11	Widerstand 120 kOhm	
R12...R14	Widerstand 3,9 kOhm	
R16	Widerstand 2,2 kOhm	
R17...R19	SIL-Widerstandsnetzwerk 8x4,7 kOhm	1 gemeinsamer Anschluß
S1...S3	Miniatur-Schiebeschalter 8-polig, DIL	
V1	Si-NPN-Transistor BC 337-25	
V2	Si-NPN-Transistor BC 547-25	
V3, V4	Si-Diode 1N4148	

R' S 1,2 u  
1,0 u

---

Anhang 4: Lötbrückentabelle

---

Auf der Video-Interface-Baugruppe befinden sich einige Lötbrücken (Jumper). Diese ermöglichen eine Anpassung der Hardware an bestimmte Erfordernisse.

- J1 - Umschaltung internes/externes ROM für den TMP  
geöffnet: externes ROM  
geschlossen: internes ROM (nur bei NS455!)  
**Bei dem hier verwendeten Typ (NS 405) immer offen!**  
Lieferzustand: geöffnet (externes ROM)
- J2 - Speichertyp-Umschaltung  
Stellung "a": 6264 (8K x 8-Bit)  
Stellung "b": 6116 (2K x 8-Bit)  
**Es darf immer nur eine Brücke ("a" oder "b") geschlossen sein!**  
Lieferzustand: Stellung "a" (6264, 8K x 8-Bit)
- J3 - Wahl des aktiven Pegels des Tastatur-STROBE-Signals  
geöffnet: H-Pegel (MFA-Standard-Tastatur)  
geschlossen: L-Pegel  
Lieferzustand: geöffnet (H-Pegel)
- J4 - Interner Signalgeber H1 ein/aus  
geöffnet: interner Signalgeber aus  
geschlossen: interner Signalgeber ein  
Lieferzustand: geschlossen (interner Signalgeber ein)
- J5 - An X3 anschließbarer externer Signalgeber ein/aus  
geöffnet: externer Signalgeber aus  
geschlossen: externer Signalgeber ein  
Lieferzustand: geschlossen (externer Signalgeber ein)

**Der in der Tastatur eingebaute Signalgeber kann nicht als externer Signalgeber verwendet werden! Als externe Signalgeber können nur piezokeramische Signalgeber verwendet werden.**

---

Anhang 4: Lötbrückentabelle

---

## J6 - Umschaltung externer/interner Reset

Stellung "a": interner Reset

Stellung "b": externer Reset

**Es darf immer nur eine Brücke ("a" oder "b") geschlossen sein!**

Lieferzustand: Stellung "b" (externer Reset)

Bei einem **externen Reset** wird das Video-Interface beim Betätigen des an der CPU-Baugruppe befindlichen Reset-Tasters zurückgesetzt. Dies kann sinnvoll sein, wenn versehentlich falsche Steuersequenzen zum Interface übertragen wurden und der Anwender nicht weiß, mit welchen Steuersequenzen er den Ursprungszustand wieder herstellen kann. In manchen Fällen kann es allerdings nachteilig sein, daß beim Rücksetzen der CPU auch das Video-Interface zurückgesetzt wird, da es dann die über die DIL-Schalter festgelegte Betriebsart einnimmt. Alle Änderungen der Betriebsart, die eventuell zuvor über Steuersequenzen vorgenommen wurden, werden damit hinfällig.

## J7 - AKD-Signal der Tastatur zum RST 6.5-Anschluß der CPU durchschleifen

geöffnet: AKD-Signal nicht durchschleifen

geschlossen: AKD-Signal durchschleifen

Lieferzustand: geschlossen (AKD-Signal auf RST 6.5-Anschluß)

Bei geschlossener Brücke erhält die CPU bei jedem Tastendruck eine Interrupt-Anforderung. Dadurch kann ein laufendes BFZ-STEUER-BASIC- oder SPS-Programm durch Betätigung einer Taste (außer SHIFT, CONTROL, BREAK) abgebrochen werden. Bei geöffneter Brücke können diese Programme nur durch die Betätigung des Reset-Tasters abgebrochen werden.

## Anhang 5: Steckerbelegungen

## X1 - MFA-Systembus (VG64-Steckerleiste)

Kontakt	Signalbezeichnung	
1 a/c	5V	Betriebsspannung
27 c	RST 6.5	AKD-Signal der Tastatur
28 c	DTR	Data Terminal Ready (Steuersignal)
29 c	TxD	Daten vom Video-Interface
30 c	RxD	Daten zum Video-Interface
31 a	+12V	Betriebsspannung
31 c	-12V	Betriebsspannung
32 a/c	0V	Masse (Ground)

## X2 - Tastatur-Anschluß (15-polige Sub-D-Buchse)

Kontakt	Signalbezeichnung	
1	DB0	Tastatur-Datenleitung 0
2	DB1	Tastatur-Datenleitung 1
3	DB2	Tastatur-Datenleitung 2
4	DB3	Tastatur-Datenleitung 3
5	DB4	Tastatur-Datenleitung 4
6	DB5	Tastatur-Datenleitung 5
7	DB6	Tastatur-Datenleitung 6
8	STB	Tastatur-STROBE
9	---	Nicht belegt
10	0V	Masse (Ground)
11	5V	Betriebsspannung der Tastatur
12	---	Nicht belegt
13	DB7	Tastatur-Datenleitung 7 (falls angeschl.)
14	AKD	Any Key Down
15	---	Nicht belegt

## X3 - Anschluß für ext. Signalgeber (2 Lötpunkte)

Kontakt	Signalbezeichnung	
1	0V	Masse (Ground)
2	BELL	Signalgeber-Ansteuerung

## X4 - Monitor-Anschluß (BNC-Buchse)

Kontakt	Signalbezeichnung	
Mitte Schirm	BAS 0V	Bild-, Austast- und Synchron-Signal Masse (Ground)

## Anhang 6: ASCII-Tabelle

## Anhang 7: Tabelle der länderspezifischen Zeichen

ASCII-Code		Nationalitäten								
hex.	dez.	USA	FRA	GER	ENG	DEN	SWE	ITA	SPA	
23	35	#	#	#	£	#	#	#	#	
24	36	\$	\$	\$	\$	\$	¤	\$	\$	
40	64	@	à	§	@	@	@	@	@	
5B	91	[	°	Ä	[	Æ	Ä	°	i	
5C	92	\	ç	Ö	\	ø	ö	\	ñ	
5D	93	]	§	Ü	]	Å	Å	é	]	
5E	94	^	^	^	^	^	Ü	^	^	
60	96	,	,	,	,	,	é	ù	,	
7B	123	{	é	ä	{	æ	ä	à	„	
7C	124		ù	ö		ø	ö	ö	ñ	
7D	125	}	è	ü	}	å	å	è	}	
7E	126	~	”	ß	~	~	ü	ì	~	

## Anhang 8: Darstellung der Steuerzeichen im Control-Mode

Im Control-Mode werden alle **Steuerzeichen**, die das Video-Interface empfängt, nicht ausgeführt, sondern als inverse Zeichen mit halber Helligkeit auf dem Bildschirm angezeigt. In der nachstehenden Tabelle sind Code und Mnemonik der Steuerzeichen sowie angezeigte Zeichen gegenübergestellt.

Zwischen den Steuerzeichen und den angezeigten Zeichen besteht folgender Zusammenhang:

$$\text{Code des angezeigten Zeichens} = \text{Code des Steuerzeichens} + 40_{\text{H}}$$

Die Codes der Steuerzeichen lassen sich auch von der Tastatur zum Mikrocomputer übertragen. Hierzu muß das in der rechten Spalte der Tabelle aufgeführte Zeichen ("angezeigtes Zeichen") bei gleichzeitiger Betätigung der CONTROL-Taste eingegeben werden.

Beispiel:

Soll z.B. das Steuerzeichen NUL (Code  $00_{\text{H}}$ ) zum Mikrocomputer übertragen werden, ist die CONTROL-Taste zusammen mit der Taste "@" zu betätigen.

Steuerzeichen		angezeigtes Zeichen <sup>1)</sup>
Code	Mnemonik	
$00_{\text{H}}$	NUL	Ø
$01_{\text{H}}$	SOH	A
$02_{\text{H}}$	STX	B
$03_{\text{H}}$	ETX	C
$04_{\text{H}}$	EOT	D
$05_{\text{H}}$	ENQ	E
$06_{\text{H}}$	ACK	F
$07_{\text{H}}$	BEL	G
$08_{\text{H}}$	BS <i>(CRSR ←)</i>	H
$09_{\text{H}}$	HT <i>(CRSR →)</i>	I
$0A_{\text{H}}$	LF <i>(CRSR ↓)</i>	J
$0B_{\text{H}}$	VT <i>(CRSR ↑)</i>	K
$0C_{\text{H}}$	FF	L
$0D_{\text{H}}$	CR	M
$0E_{\text{H}}$	SO	N
$0F_{\text{H}}$	SI	O

Steuerzeichen		angezeigtes Zeichen <sup>1)</sup>
Code	Mnemonik	
$10_{\text{H}}$	DLE	P
$11_{\text{H}}$	DC1	Q
$12_{\text{H}}$	DC2	R
$13_{\text{H}}$	DC3	S
$14_{\text{H}}$	DC4	T
$15_{\text{H}}$	NAK	U
$16_{\text{H}}$	SYN	V
$17_{\text{H}}$	ETB	W
$18_{\text{H}}$	CAN	X
$19_{\text{H}}$	EM	Y
$1A_{\text{H}}$	SUB	Z
$1B_{\text{H}}$	ESC	[
$1C_{\text{H}}$	FS	\
$1D_{\text{H}}$	GS	]
$1E_{\text{H}}$	RS	^
$1F_{\text{H}}$	US	-

<sup>1)</sup> Die angezeigten Zeichen sind abhängig vom eingestellten länderspezifischen Zeichensatz. Für diese Tabelle wurde der USA-Zeichensatz (empfohlene Grundeinstellung und Tastaturschriftung) verwendet.

---

Anhang 9: Bedeutung der Schalterstellungen der Schalter-Reihen S1 bis S3

---

Auf der Baugruppe befinden sich drei Schalter-Reihen. Jede dieser Schalter-Reihen besteht aus acht Schaltern. Die einzelnen Schalter-Reihen sind mit S1, S2 und S3 bezeichnet. S1.1 bezeichnet den Schalter Nummer 1 in der Schalter-Reihe S1. Die anderen Schalter sind entsprechend nummeriert. Mit Hilfe dieser Schalter kann die Grundeinstellung des Video-Interfaces vorgenommen werden. Diese Grundeinstellung wird nach dem Einschalten des Interfaces aktiv. Die Grundeinstellung kann aber während des Betriebes durch die Übertragung entsprechender Steuerzeichen bzw. durch das Set-Up-Menü verändert werden. Die folgende Übersicht zeigt die Einstellmöglichkeiten dieser Schalter-Reihen:

- Schalter-Reihe S1:

S1.1 (Emulations-Mode):

S1.1	Emulations-Mode
OFF	TVI 950
ON	MAT 85

S1.2 (Cursor-Art):

S1.2	Cursor-Art
OFF	blinkend
ON	ruhend

S1.3 (Cursor-Form):

S1.3	Cursor-Form
OFF	Strich
ON	Block

S1.4 (Bildhintergrund):

S1.4	Bildhintergrund
OFF	dunkel
ON	hell

S1.5 (Reserviert, muß immer in Stellung OFF sein)

## Anhang 9: Bedeutung der Schalterstellungen der Schalter-Reihen S1 bis S3

S1.6 (Handshake):

S1.6	Handshake
OFF	Hardware (DTR)
ON	Software (XON/XOFF)

S1.7 (Reserviert, muß immer in Stellung OFF sein)

S1.8 (Statuszeile):

S1.8	Statuszeile
OFF	aus
ON	ein

## - Schalter-Reihe S2

S2.1 (Wortlänge):

S2.1	Wortlänge
OFF	7 Bit
ON	8 Bit

S2.2 (Zeichensatz):

S2.3

S2.4

S2.2	S2.3	S2.4	Zeichensatz
OFF	OFF	OFF	USA
ON	OFF	OFF	Frankreich
OFF	ON	OFF	Deutschland
ON	ON	OFF	England
OFF	OFF	ON	Dänemark
ON	OFF	ON	Schweden
OFF	ON	ON	Italien
ON	ON	ON	Spanien

**Anhang 9: Bedeutung der Schalterstellungen der Schalter-Reihen S1 bis S3**

S2.5 (Zeilenanzahl):

S2.6

S2.5	S2.6	Zeilenanzahl
OFF	OFF	22
ON	OFF	24
OFF	ON	26
ON	ON	28

S2.7 (Spaltenanzahl):

S2.8

S2.7	S2.8	Spaltenanzahl
OFF	OFF	72
ON	OFF	80
OFF	ON	88
ON	ON	96

**- Schalter-Reihe S3**

S3.1 (Anzahl der Stopbits):

S3.1	Stopbits
OFF	1 Bit
ON	2 Bit

S3.2 (Paritätsbit):

S3.3

S3.4

S3.2	S3.3	S3.4	Paritätsbit
OFF	OFF	OFF	kein Paritätsbit, no Parity
ON	OFF	OFF	ungerade Parität, odd Parity
OFF	ON	OFF	kein Paritätsbit, no Parity
ON	ON	OFF	gerade Parität, even Parity
OFF	OFF	ON	kein Paritätsbit, no Parity
ON	OFF	ON	immer "1", Mark Parity
OFF	ON	ON	kein Paritätsbit, no Parity
ON	ON	ON	immer "0", Space Parity

---

Anhang 9: Bedeutung der Schalterstellungen der Schalter-Reihen S1 bis S3

---

S3.5 (Baud-Rate):

S3.6

S3.7

S3.8

S3.5	S3.6	S3.7	S3.8	Baud-Rate
OFF	OFF	OFF	OFF	110 Bd
ON	OFF	OFF	OFF	134,5 Bd
OFF	ON	OFF	OFF	150 Bd
ON	ON	OFF	OFF	300 Bd
OFF	OFF	ON	OFF	600 Bd
ON	OFF	ON	OFF	1200 Bd
OFF	ON	ON	OFF	1800 Bd
ON	ON	ON	OFF	2400 Bd
OFF	OFF	OFF	ON	3600 Bd
ON	OFF	OFF	ON	4800 Bd
OFF	ON	OFF	ON	7200 Bd
ON	ON	OFF	ON	9600 Bd
OFF	OFF	ON	ON	19200 Bd
ON	OFF	ON	ON	19200 Bd
OFF	ON	ON	ON	19200 Bd
ON	ON	ON	ON	9600 Bd

---

 Anhang 10: Tabelle der Attribut-Bytes für die Escape-Sequenz "ESC G"
 

---

Mit Hilfe der Escape-Sequenz "ESC G" können den einzelnen Zeichen im TVI 950-Mode des Interfaces verschiedene Attribute zugewiesen werden. Hierzu ist nach den beiden Bytes für die einleitenden Zeichen "ESC G" ein drittes Byte zu übertragen. Mit diesem dritten Byte "Attribute+30<sub>H</sub>" werden die gewünschten Attribute festgelegt. Die Tabelle auf der nächsten Seite soll helfen, dieses Byte zu ermitteln.

Im rechten Tabellenteil findet man die einzelnen Attribute: "unsichtbar", "blinkend", "invers", "unterstrichen", "doppelt breit" und "doppelt hoch". Aktive Attribute sind in der entsprechenden Spalte durch das Symbol "\*" gekennzeichnet. Zur Ermittlung des dritten Bytes der Escape-Sequenz sucht man die Tabellen-Zeile auf, in der die gewünschte Attribut-Kombination durch "\*" -Symbole gekennzeichnet ist. Im linken Tabellenteil findet man das entsprechende Byte (jeweils in ASCII-, Hexadezimal- und Dezimalcodierung). Die Tabelle enthält alle zulässigen Attribut-Kombinationen.

Beispiel:

Es soll die Attribut-Kombination "unterstrichen" / "blinkend" eingeschaltet werden. In der Tabelle ist hierfür folgende Zeile gültig:

3. Byte "Attribute+30 <sub>H</sub> "			Bits					
ASCII <sup>1)</sup>	hex.	dez.	5 doppelt hoch	4 doppelt breit	3 unterstr.	2 invers	1 blinkend	0 unsichtbar
:	3A	58	-	-	*	-	*	-

Aus der Tabelle entnimmt man als drittes Byte der Escape-Sequenz das ASCII-Zeichen ":". Hiermit ergibt sich folgende Gesamtsequenz:

ESC G : (ASCII codiert)  
 1B<sub>H</sub> 47<sub>H</sub> 3A<sub>H</sub> (hex. codiert)  
 27<sub>D</sub> 71<sub>D</sub> 58<sub>D</sub> (dez. codiert)

## Anhang 10: Tabelle der Attribut-Bytes für die Escape-Sequenz "ESC G"

3. Byte "Attribute+30 <sub>H</sub> "			Bits					
ASCII <sup>1)</sup>	hex.	dez.	5 doppelt hoch	4 doppelt breit	3 unterstr.	2 invers	1 blinkend	0 unsichtbar
0	30	48	-	-	-	-	-	-
1	31	49	-	-	-	-	-	*
2	32	50	-	-	-	-	*	-
3	33	51	-	-	-	-	*	*
4	34	52	-	-	-	*	-	-
5	35	53	-	-	-	*	-	*
6	36	54	-	-	-	*	*	-
7	37	55	-	-	-	*	*	*
8	38	56	-	-	*	-	-	-
9	39	57	-	-	*	-	-	*
:	3A	58	-	-	*	-	*	-
;	3B	59	-	-	*	-	*	*
<	3C	60	-	-	*	*	-	-
=	3D	61	-	-	*	*	-	*
>	3E	62	-	-	*	*	*	-
?	3F	63	-	-	*	*	*	*
Ø	40	64	-	*	-	-	-	-
A	41	65	-	*	-	-	-	*
B	42	66	-	*	-	-	*	-
C	43	67	-	*	-	-	*	*
D	44	68	-	*	-	*	-	-
E	45	69	-	*	-	*	-	*
F	46	70	-	*	-	*	*	-
G	47	71	-	*	-	*	*	*
H	48	72	-	*	*	-	-	*
I	49	73	-	*	*	-	-	*
J	4A	74	-	*	*	-	*	-
K	4B	75	-	*	*	-	*	*
L	4C	76	-	*	*	*	-	-
M	4D	77	-	*	*	*	-	*
N	4E	78	-	*	*	*	*	-
O	4F	79	-	*	*	*	*	*
P	50	80	*	-	-	-	-	-
R	52	82	*	-	-	-	*	-
T	54	84	*	-	-	*	-	-
V	56	86	*	-	-	*	*	-
X	58	88	*	-	*	-	-	-
Z	5A	90	*	-	*	-	*	-
\	5C	92	*	-	*	-	-	-
^	5E	94	*	-	*	-	-	-
,	60	96	*	*	-	-	-	-
b	62	98	*	*	-	-	*	-
d	64	100	*	*	-	*	-	-
f	66	102	*	*	-	*	*	-
h	68	104	*	*	*	-	-	-
j	6A	106	*	*	*	-	*	-
l	6C	108	*	*	*	*	*	-
n	6E	110	*	*	*	*	*	-

<sup>1)</sup> Für die Darstellung der ASCII-Zeichen wurde der USA-Zeichensatz (empfohlene Grundeinstellung und Tastaturschriftung) verwendet.

## Anhang 11: Steuerzeichen im MAT 85-Mode

## - 1-Byte-Steuerzeichen

Funktion	Bezeichnung	hex. Code	dez. Code	Seite
Bell	BEL	07	7	15
Cursor nach links (Backspace)	BS	08	8	15
Cursor nach rechts	HT	09	9	15
Zeilenvorschub (Cursor nach unten)	LF	0A	10	16
Cursor nach oben	VT	0B	11	16
Bildschirm löschen, Cursor nach oben links	FF	0C	12	16
Wagenrücklauf, Zeilenrest löschen	CR	0D	13	16
Zeile löschen	SUB	1A	26	16
Cursor nach unten	ESC	1B	27	17
Cursor nach oben links	FS	1C	28	17
Cursor an den Zeilenanfang	GS	1D	29	17

## - DLE-Sequenzen

Funktion	Bezeichnung	hex. Code	dez. Code	Seite
Modeumschaltung (TVI 950)	DLE DLE 1	10 10 31	16 16 49	17
Modeumschaltung (MAT 85)	DLE DLE 2	10 10 32	16 16 50	18
Modeabfrage	DLE DLE ?	10 10 3F	16 16 63	18
Software-Reset	DLE DLE @	10 10 40	16 16 64	18
Versionsabfrage	DLE DLE V	10 10 56	16 16 86	18

## Anhang 12: Steuerzeichen im TVI 950-Mode

## - 1-Byte-Steuerzeichen

Funktion	Bezeichnung	hex. Code	dez. Code	Seite
Bell	BEL	07	7	19
Cursor nach links (Backspace)	BS	08	8	19
Horizontal-Tabulator	HT	09	9	20
Zeilenvorschub (Cursor nach unten)	LF	0A	10	20
Cursor nach oben	VT	0B	11	20
Cursor nach rechts	FF	0C	12	20
Wagenrücklauf (Cursor an den Zeilenanfang)	CR	0D	13	21
Cursor nach unten	SYN	16	22	21
Bildschirm löschen, Cursor nach oben links	SUB	1A	26	21
Cursor nach oben links	RS	1E	30	21

## - DLE-Sequenzen

Funktion	Bezeichnung	hex. Code	dez. Code	Seite
Modeumschaltung (TVI 950)	DLE DLE 1	10 10 31	16 16 49	31
Modeumschaltung (MAT 85)	DLE DLE 2	10 10 32	16 16 50	31
Modeabfrage	DLE DLE ?	10 10 3F	16 16 63	31
Software-Reset	DLE DLE @	10 10 40	16 16 64	31
Versionsabfrage	DLE DLE V	10 10 56	16 16 86	32

## Anhang 12: Steuerzeichen im TVI 950-Mode

## - Escape-Sequenzen (nach Funktionen geordnet)

Erfordert eine Escape-Sequenz zusätzliche Parameter, so ist hier der Platzhalter " \_\_ " angegeben.

Funktion	Bezeichnung	hex. Code	dez. Code	Seite
<u>Bildschirm löschen, Cursor nach oben links</u>				
Bildschirm löschen, Cursor nach oben links	ESC *	1B 2A	27 42	22
Bildschirm löschen, Cursor nach oben links	ESC +	1B 2B	27 43	22
Bildschirm löschen, Cursor nach oben links	ESC ,	1B 2C	27 44	22
Bildschirm löschen, Cursor nach oben links	ESC :	1B 3A	27 58	22
Bildschirm löschen, Cursor nach oben links	ESC ;	1B 3B	27 59	22
<u>Schriftfeld editieren</u>				
Leerzeichen einfügen	ESC Q	1B 51	27 81	22
Zeichen löschen	ESC W	1B 57	27 87	22
Zeile einfügen	ESC E	1B 45	27 69	22
Zeile löschen	ESC R	1B 52	27 82	23
Zeile ab Cursor-Position löschen	ESC T	1B 54	27 84	23
Zeile ab Cursor-Position löschen	ESC t	1B 74	27 116	23
Schriftfeld ab Cursor-Position löschen	ESC Y	1B 59	27 89	23
Schriftfeld ab Cursor-Position löschen	ESC y	1B 79	27 121	23
<u>Cursor-Positionierung</u>				
Cursor-Position setzen	ESC = __	1B 3D __	27 61 __	23
Cursor-Position abfragen	ESC ?	1B 3F	27 63	24
Cursor nach links zum nächsten Tabulator (Backtab)	ESC I	1B 49	27 73	25
<u>Zeichensätze, Zeichen- und Cursor-Darstellung</u>				
Zeichensatz wählen	ESC z _	1B 7A _	27 122 _	25
Attribute setzen/löschen	ESC G _	1B 47 _	27 71 _	27
Halbe Helligkeit	ESC >	1B 29	27 41	29
Volle Helligkeit	ESC (	1B 28	27 40	29
Bildhintergrund hell	ESC b	1B 62	27 98	29
Bildhintergrund dunkel	ESC d	1B 64	27 100	29
Control-Mode ein	ESC U	1B 55	27 85	29
Cursor-Darstellung	ESC . _	1B 2E _	27 46 _	30

## Anhang 12: Steuerzeichen im TVI 950-Mode

## - Escape-Sequenzen (nach Codes geordnet)

Erfordert eine Escape-Sequenz zusätzliche Parameter, so ist hier der Platzhalter "\_" angegeben.

Funktion	Bezeichnung	hex. Code	dez. Code	Seite
Volle Helligkeit	ESC (	1B 28	27 40	29
Halbe Helligkeit	ESC )	1B 29	27 41	29
Bildschirm löschen, Cursor nach oben links	ESC *	1B 2A	27 42	22
Bildschirm löschen, Cursor nach oben links	ESC +	1B 2B	27 43	22
Bildschirm löschen, Cursor nach oben links	ESC ,	1B 2C	27 44	22
Cursor-Darstellung	ESC . <u>_</u>	1B 2E <u>_</u>	27 46 <u>_</u>	30
Bildschirm löschen, Cursor nach oben links	ESC :	1B 3A	27 58	22
Bildschirm löschen, Cursor nach oben links	ESC ;	1B 3B	27 59	22
Cursor-Position setzen	ESC = <u>_</u> <u>_</u>	1B 3D <u>_</u> <u>_</u>	27 61 <u>_</u> <u>_</u>	23
Cursor-Position abfragen	ESC ?	1B 3F	27 63	24
Zeile einfügen	ESC E	1B 45	27 69	22
Attribute setzen/löschen	ESC G <u>_</u>	1B 47 <u>_</u>	27 71 <u>_</u>	27
Cursor nach links zum nächsten Tabulator (Backtab)	ESC I	1B 49	27 73	25
Leerzeichen einfügen	ESC Q	1B 51	27 81	22
Zeile löschen	ESC R	1B 52	27 82	23
Zeile ab Cursor-Position löschen	ESC T	1B 54	27 84	23
Control-Mode ein	ESC U	1B 55	27 85	29
Zeichen löschen	ESC W	1B 57	27 87	22
Schriftfeld ab Cursor-Position löschen	ESC Y	1B 59	27 89	23
Bildhintergrund hell	ESC b	1B 62	27 98	29
Bildhintergrund dunkel	ESC d	1B 64	27 100	29
Zeile ab Cursor-Position löschen	ESC t	1B 74	27 116	23
Schriftfeld ab Cursor-Position löschen	ESC y	1B 79	27 121	23
Zeichensatz wählen	ESC z <u>_</u>	1B 7A <u>_</u>	27 122 <u>_</u>	25