

# GRIP-6

## Board-Grafikterminal

### Befehlssatz

Copyright © CONITEC 1989, 1990, 7/1991

Hardware: Johannes C. Lotter  
Layout: Gesa Emde  
Software: Bernhard Emese / Thorsten Siebeking  
Handbuch: Johannes C. Lotter

Kein Teil dieser Dokumentation darf ohne ausdrückliche schriftliche Genehmigung von CONITEC reproduziert, vervielfältigt, gespeichert oder übersetzt werden. Für die Richtigkeit der hier angegebenen Daten übernehmen wir keine Haftung. Änderungen, die dem Fortschritt dienen, behalten wir uns vor, auch ohne dies besonders anzukündigen.

 **CONITEC DATENSYSTEME**

6100 Darmstadt • Grafenstr. 31 • ☎ (06151) 26013 • Fax (06151) 26015

## **GRIP-Befehlssatz - Inhaltsübersicht**

1. Text-Modus (TVI950/VT100-Emulation)
  - 1.1 TVI-Modus
  - 1.2 VT100-Modus
2. Grafik-Modus (Tektronix-Emulation)
  - 2.1 Tektronix-Steuercodes
  - 2.2 Tektronix-Escape-Sequenzen
  - 2.3 Vektor-, Punkt- und Inkrementalmodus
3. Farbe
  - 3.1 Farbindizes
  - 3.2 Farbdarstellung und Look-Up-Tafel
4. Tastatur
5. Transferbefehle
  - 5.2 Centronics-Schnittstelle
  - 5.3 ECB-Bus-Schnittstelle
  - 5.4 RS232/RS422-Schnittstelle
  - 5.5 Bildspeicher-Transfer
  - 5.6 Zeichensätze
  - 5.7 Userprogramm/Portzugriffe
  - 5.8 Tasten-Umcodetabelle
  - 5.9 Soundgenerator
  - 5.10. Statuszeilen und Uhr
  - 5.11. Systempatch
  - 5.12. Hardcopy
6. Bildschirmseiten und Spooler
7. Lichtgriffel- und Statusabfrage
9. Bildschirmformate
10. Genlock und Bildverarbeitung
12. Set-Up-Menu

GRIP versteht Befehle eines TVI950-, VT100/VT220- oder Tektronix 4010-Grafikterminals. Zur Steuerung interner Funktionen dienen die *Dopplescape-Sequenzen*, die durch zwei Escape-Zeichen (1Bh) eingeleitet werden.

Manche Befehle sind nur für GRIP-6B, andere nur für GRIP-6C implementiert. Dies ist bei dem Befehl jeweils angegeben. Die Kürzel für die Befehls-Parameter haben folgende Bedeutung (nachgestelltes 'h' kennzeichnet Hexadezimaldarstellung):

x,y,z = Byte (00h-FFh)  
 xx,yy = 16-Bit-Wort (LSB zuerst), z.b. 0Ah 10h = 100Ah = 4106  
 a,b,c = ASCII-Zeichen (20h-7Fh), z.b. '1' (31h)  
 aa,bb = ASCII-Hex-Byte (2 Zeichen), z.b. '41' (34h 31h)  
 aaaa = ASCII-Hex-Wort (4 Zeichen), z.b. '100A' (31h 30h 30h 41h)  
 Pn = ASCII-Dezimal-Zahl, z.b. '666' (36h 36h 36h)  
  
 <ESC> = 1Bh  
 <CR> = 0Dh  
 ^... = Controlzeichen, z.b. ^C = Control-C = 03h

Bei manchen Parameter-Bytes sind die einzelnen Bits in der Form

Bit7 Bit6 Bit5 Bit4 Bit3 Bit2 Bit1 Bit0

dargestellt, wenn sie einzeln unterschiedliche Bedeutung haben. Ein vorangestellter Schrägstrich (/) zeigt an, daß die folgenden Zeichen als Antwort von GRIP zum Host gesendet werden. Dabei ist das 8. Bit der Antwort-Bytes gesetzt, um sie von normalen Tastaturcodes zu unterscheiden.

In der Kommando-Übersicht sind einige Parameter mit der Bemerkung (default) versehen. Die durch diese Befehle eingestellten Parameter sind beim Einschalten oder Rücksetzen der Karte ausgewählt; ebenso, wenn der folgende Befehl gegeben wurde:

**ESC ESC 'R'**      **Reset-Befehl: Bildschirm und alle Puffer löschen, alle Funktionen auf die Default-Werte zurücksetzen.**

Der Reset-Befehl löscht als einziger Befehl auch den Hostpuffer! Deswegen werden die unmittelbar folgenden Befehle, wenn sie schon im Puffer stehen, nicht mehr ausgeführt. Es empfiehlt sich in den Programmablauf nach Senden des Reset-Befehls vor dem Absenden weiterer Befehle eine Verzögerung (ca. 500 ms) einzubauen.

## 1 Der Text-Modus

Im Text-Betrieb, der nach Reset aktiv ist, wird von der GRIP ein Televideo-950- oder ein VT100-Terminal simuliert. Die Simulation umfaßt die Bildschirm-Editbefehle (Zeilenlöschen, -einfügen, Cursorpositionierung usw.), einige Sonderbefehle wie z.B. Cursor-Umdefinition sowie die terminalspezifische Strichgrafik. Damit lassen sich die Installationsmenues von Softwarepaketen wie WORDSTAR, MULTIPLAN usw. benutzen.

GRIP schaltet automatisch von der TVI- in die VT100-Emulation, wenn ein VT100-spezifischer Befehl (ESC '['..., ESC '#'..., <ESC> 'c') gesendet wird (der Befehl wird dann ausgeführt!). Zum Zurückschalten in die TVI-Emulation dient der TVI-Befehl ESC '+' (Bildschirmlöschen).

Der TVI-Modus wird mit folgenden Dopplescape-Sequenzen aktiviert:

ESC ESC 'A'	Text-Modus einschalten (default).
ESC '+'	Bildschirm löschen, TVI-Emulation aktivieren.

Der VT100-Modus wird mit folgenden Dopplescape-Sequenzen aktiviert:

ESC ESC 'A'	Text-Modus einschalten (default).
ESC 'c'	Bildschirm initialisieren, VT100-Emulation aktivieren.

**1.1: TVI-Steuercodes**

- ^H (08h)** Cursor nach links (Backspace). Steht der Cursor bereits am Zeilenanfang, so springt er in die letzte Spalte der darüberliegenden Zeile. In der Home-Position (Zeile 0, Spalte 0) hat dieser Code keine Wirkung.
- ^I (09h)** Cursor nach rechts. Am Zeilenende springt der Cursor auf den Anfang der nächsten Zeile. War die Zeile bereits die letzte auf dem Bildschirm, so rollt das Bild um eine Zeile nach oben (Zeilenvorschub).
- ^K (0Bh)** Cursor auf. In der ersten Zeile am oberen Bildschirmrand hat dieser Code keine Wirkung.
- ^V (16h)** Cursor ab. In der letzten Zeile erfolgt ein Zeilenvorschub, d.h. das Bild rollt um eine Zeile nach oben.
- ^J (0Ah)** LINE FEED: Zeilenvorschub, wie <^V>.
- ^M (0Dh)** CARRIAGE RETURN: Wagenrücklauf. Setzt den Cursor an den Zeilenanfang.
- ^\_ (0Fh)** Neue Zeile. Entspricht einem Wagenrücklauf plus Zeilenvorschub und setzt den Cursor an den Anfang der nächsten Zeile.
- ^^ (1Eh)** Cursor in die Home-Position (Zeile 0, Spalte 0 oben links).
- ^Z (1Ah)** Bildschirm löschen.

- ^L (0Ch)** FORM FEED: Bildschirm löschen, Cursor in die Home-Position (wie <^^> und <^Z> nacheinander). Dieser Code wird für CP/M 3.0 benötigt.
- ^G (07h)** BEL: Glockensignal. Hier wird der Inhalt des Soundpuffers (s.u.) abgespielt; wenn dort nichts steht, ertönt stattdessen ein kurzes Klingelsignal.

## 1.2: TVI-Escape-Sequenzen

**ESC '+'** Löschen wie <^L>. Aktiviert TVI-Emulation.

**ESC '/'** Löschen wie <^L>.

**ESC ' ' a** Cursorattribute setzen.

- a = '0': Unsichtbarer Cursor  
 '1': Blinkender Block (def)  
 '2': Stehender Block  
 '3': Blinkende Linie  
 '4': Stehende Linie

**ESC '= ' a b** Cursorposition setzen.

**ESC '? ' /a b <CR>** Cursorposition abfragen.

a = Zeilennummer + 20h

b = Spaltennummer + 20h

Beispiel: ESC ' = A' (1Bh 3Dh 20h 41h) setzt den Cursor auf Zeile 0 (oberste Zeile), Spalte 33.

Beim Abfrage-Befehl ist das 8. Bit der drei Antwort-Bytes (a,b,<CR>) gesetzt, um die Positionszeichen von normalen Tastaturbytes zu unterscheiden.

- ESC '1'** Bildschirm um eine Zeile nach oben rollen.
- ESC '2'** Bildschirm um eine Spalte nach rechts rollen.
- ESC '3'** Bildschirm um eine Zeile nach unten rollen.
- ESC '4'** Bildschirm um eine Spalte nach links rollen.

Der Cursor bewegt sich mit, bleibt also auf dem Zeichen, auf dem er steht, bis er an den Rand stößt. Die Statuszeilen bewegen sich nicht mit.

- ESC 'E'** Leerzeile bei Cursorposition einfügen. Alle folgenden Zeilen rollen nach unten, der Cursor springt zum Anfang der neu eingefügten Zeile.
- ESC 'R'** Zeile bei Cursorposition löschen. Alle folgenden Zeilen rollen nach oben. Der Cursor springt zum Zeilenanfang.
- ESC 'T'** Zeile ab einschl. Cursorposition bis zum Ende löschen.
- ESC 'Y'** Seite ab einschl. Cursorposition bis zum Ende löschen.

ESC 'G' x Zeichenattribute setzen.

x = 0 b s1 s0 u i d v

b = Breitschrift

s1-0 = 00: Index tief (Subscript)

01: Indexschrift

10: Index hoch (Superscript)

11: Normale Schrift (def)

u = Unterstreichen

i = Invertieren

d = Durchstreichen

v = Unsichtbare Schrift

Mit den beiden Schriftarten Normal- und Indexschrift kann man zwischen zwei Zeichensätzen hin- und herschalten; der Index-Zeichensatz läßt sich zusätzlich noch hoch- und tiefgestellt darstellen. Beide Zeichensätze lassen sich vom Host umprogrammieren (s.u.), so daß anstelle der Indexschrift z.b. auch kyrillische Schrift o.ä. stehen könnte.

Zu jedem Zeichen lassen sich fünf unabhängige weitere Attribute auswählen: Breitschrift (jedes Zeichen wird in doppelter Breite dargestellt), Invertieren (schwarzes Zeichen auf weißem Feld), durchstreichen, unterstreichen und unsichtbar. Die Attribute lassen sich beliebig kombinieren, sind jedoch mit einer Reduzierung der Übertragungsgeschwindigkeit verbunden.

Außer den angegebenen Attributen lassen sich noch Vorder- und Hintergrundfarben sowie Blinken einstellen (s. Kapitel "Farbe").

Beispiel: ESC 'G|' (1Bh 47h 5Ch) schaltet unterstrichene, breite, invertierte Indexschrift ein.

ESC ']' Invertieren ein.

ESC '[' Invertieren aus (def).

Diese Befehle lassen die übrigen Attribute unverändert und werden für die Invertierungs-Option von WORDSTAR, TURBO-PASCAL etc. verwendet.

ESC 'n' Bildschirm ein (def).

ESC 'o' Bildschirm dunkel.

Diese Befehle können benutzt werden, um längere Textänderungen, Scrollen u.a. für den Benutzer unsichtbar zu machen. Nach erfolgtem Bildaufbau wird der Bildschirm mit ESC 'n' dann wieder eingeschaltet.

ESC '\$' TVI-Strichgrafik ein.

ESC '%' TVI-Strichgrafik aus (def).

Die 26 Strichgrafik-Zeichen sind nach ESC '\$' anstelle der Großbuchstaben 'A'-'Z' verfügbar; sie sind für Spreadsheet-Programme wie z.B. MULTIPLAN gedacht und enthalten Feldumrandungen, Blöcke, Graufelder und verschiedene Muster.

### 1.2.1. VT100-Steuercodes

**NUL (00h)** ignoriert

**BEL (07h)** Summer

**BS (08h)** Cursor nach links

**HT (09h)** Cursor nächster Tab

**LF (0Ah)** Cursor nach unten, Scrollen in letzter Zeile

**VT (0Bh)** wie LF

**FF (0Ch)** wie LF

**CR (0Dh)** Cursor an Zeilenanfang

**SO (0Eh)** Zeichensatz 1

**SI (0Fh)** Zeichensatz 2

**DC1 (11h)** XON

**DC3 (13h)** XOFF

**CAN (18h)** Steuersequenzabbruch

**SUB (1Ah)** wie CAN

**ESC (1Bh)** Steuersequenzstart

**DEL (7Fh)** ignoriert

### 1.2.2. VT100-Escape-Sequenzen

Im Gegensatz zum TVI-Modus liegt die Home-Position (erste Zeile und Spalte) nicht bei (0,0), sondern bei (1,1).

#### Relative Cursorbewegung

ESC '[' *P<sub>n</sub>* 'A' CUU Cursor um *P<sub>n</sub>* nach oben  
 ESC '[' *P<sub>n</sub>* 'B' CUD Cursor um *P<sub>n</sub>* nach unten  
 ESC '[' *P<sub>n</sub>* 'C' CUF Cursor um *P<sub>n</sub>* nach rechts  
 ESC '[' *P<sub>n</sub>* 'D' CUB Cursor um *P<sub>n</sub>* nach links

ESC '[A' CUU Cursor um 1 nach oben  
 ESC '[B' CUD Cursor um 1 nach unten  
 ESC '[C' CUF Cursor um 1 nach rechts  
 ESC '[D' CUB Cursor um 1 nach links

#### Absolute Cursoradressierung

ESC '[' *P<sub>l</sub>* ';' *P<sub>v</sub>* 'H' CUP Cursor auf Zeile *P<sub>l</sub>*, Spalte *P<sub>v</sub>*  
 ESC '[' *P<sub>l</sub>* ';' *P<sub>v</sub>* 'f' CUP wie vor (Zeilen, Spalten beginnen mit 1;1)

#### Zeilenweise Cursorbewegung

ESC 'D' IND wie LF  
 ESC 'M' RIN Cursor um eine Zeile nach oben  
 ESC 'E' wie CR + LF

#### Cursorspeicher

ESC '7' DECSC Cursor und Attribute speichern  
 ESC '8' DECRC Cursor und Attribute wiederherstellen

## Löschbefehle

ESC '[0K'	Lösche von Cursor bis Zeilenende
ESC '[K'	wie vor
ESC '[1K'	Lösche Zeilenanfang bis Cursor
ESC '[2K'	Lösche Zeile (mit Leerzeichen)
ESC '[0J'	Lösche Bildrest ab Cursor
ESC '[J'	wie vor
ESC '[1J'	Lösche Bildanfang bis Cursor
ESC '[2J'	Lösche Bild

## Zeilenbefehle (ähnl. VT131).

ESC '[ Pn 'L'	<i>Pn</i> Zeilen an Cursorposition einfügen.
ESC '[ Pn 'M'	<i>Pn</i> Zeilen ab Cursorposition ausfügen.

## Scroll-Befehle

ESC '[ Pn 'T'	Bild um <i>Pn</i> Zeilen nach oben rollen.
ESC '[ Pn 'S'	Bild um <i>Pn</i> Zeilen nach unten rollen.

## Scroll-Bereich

ESC '[ P <sub>t</sub> ' ; P <sub>b</sub> 'r'	Nur rollen im Bereich von Zeile <i>P<sub>t</sub></i> (oben) bis <i>P<sub>b</sub></i> (unten).
--	---

## Terminal-Modi

ESC '[?' '25' 'l'	Cursor aus
ESC '[?' '25' 'h'	Cursor ein
ESC '[?' '7' 'l'	Auto-Wrap aus
ESC '[?' '7' 'h'	Auto-Wrap ein

**Zeilenattribute**

<b>ESC '#3'</b>	Obere Zeilenhälfte bei doppelter Zeichenhöhe
<b>ESC '#4'</b>	Untere Zeilenhälfte bei doppelter Zeichenhöhe
<b>ESC '#5'</b>	Zeile Normal
<b>ESC '#6'</b>	Zeile Breitschrift, einfach hoch

**Tabulatoren**

<b>ESC 'H'</b>	TAB setzen bei Cursor
<b>ESC '{0g'</b>	TAB löschen bei Cursor
<b>ESC '{3g'</b>	Lösche alle TABs

## Zeichenattribute

ESC '[' *Ps* ']'... *Ps* 'm'      Zeichenattribute einstellen

<i>Ps</i> =	leer	alle Attribute aus
'0'		alle Attribute aus
'1'		wie <i>Ps</i> = '37'
'4'		Unterstreichen
'5'		Blinken
'7'		Zeichen invers
'22'		wie <i>Ps</i> = '39'
'24'		Nicht unterstreichen
'25'		Nicht blinken
'27'		Nicht invers
'30'		Graue Schrift
'31'		Rote Schrift
'32'		Grüne Schrift
'33'		Gelbe Schrift
'34'		Blaue Schrift
'35'		Magenta Schrift
'36'		Cyan Schrift
'37'		Weißer Schrift
'39'		Default-Schrift (EEPROM)
'40'		Schwarzer Hintergrund
'41'		Roter Hintergrund
'42'		Grüner Hintergrund
'43'		Gelber Hintergrund
'44'		Blauer Hintergrund
'45'		Magenta Hintergrund
'46'		Cyan Hintergrund
'47'		Grauer Hintergrund
'49'		Default-Hintergrund (EEPROM)

Bei GRIP-6C lassen sich die Farbtöne per Set-Up-Menu (s.u.) beliebig einstellen. Bei GRIP-6 ist das Blinken mit den Farben gekoppelt; bei Wahl des Attributes '5' blinkt die augenblicklich eingestellte Schriftfarbe.

## Zeichensätze

ESC 'a	Zeichensatz wählen
a = 'A' (41h)	UK ASCII Zeichensatz
'B'	US ASCII
'R'	Französisch
'f'	Französisch
'K'	Deutsch
'S'	Spanisch
'G'	Schwedisch
' ' (20h)	Dänisch/Norwegisch
'I'	Italienisch
'O' (30h)	Strichgrafik

## Sonstiges

ESC 'c'      Bildschirm Reset (Kaltstart)

ESC '[Ps n'    Status Report

Ps = '5'      Sende Status

/ ESC '[0n' : Ok

/ ESC '[3n' : Fehler

Ps = '6'      Sende Cursor Position

/ ESC '[Pl 'R' : Pl = Zeile, Pc = Spalte

## 2

**Der Tektronix-Modus**

Während der Text-Modus für die Textverarbeitung gedacht ist, dient der Tektronix-Modus der Darstellung von Grafiken oder Schaubildern. Er wird mit der folgenden Doppelescape-Sequenz aktiviert:

<ESC><ESC> 'T' Tektronix-Modus einschalten.

Das Tektronix-Terminal kann selbst in unterschiedlichen Betriebsarten arbeiten. Im **Alpha-Modus** verhält es sich wie ein "abgemagertes" Text-Terminal. Der Cursor ist dabei unsichtbar, und Editieren ist nur eingeschränkt möglich. Dieser Modus wird zum Betiteln von Grafiken verwendet.

Im **Punkt-, Vektor- oder Block-Modus** zeichnet GRIP Punkte, Linien oder Rechtecke zwischen vorgegebenen Bildschirmkoordinaten. Im **Inkremental-Modus** wird ein mechanischer Plotter simuliert; es können kleinere Figuren aus Einzelpunkten zusammengesetzt werden. Es sind einige Befehle implementiert, die über den Original-Tektronix-Befehlssatz hinausgehen, z.b. Löschen, Invertieren, 'Replace', Spray, Block-, Dreieck- und Flächenfüllen.

Wird im Tektronix-Alpha-Modus ein Zeichen auf eine Stelle geschrieben, auf der bereits eins steht, so werden normalerweise beide Zeichen übereinander dargestellt; in der 'Replace'-Darstellung wird hingegen wie im Text-Modus das "ältere" Zeichen gelöscht. Zeichen können vertikal auf beliebige Pixelposition plaziert werden.

### 2.1: Tektronix-Control-Codes

- ^H (08h) Alpha-Cursor um 8 Pixel nach links.
- ^I (09h) Alpha-Cursor um 8 Pixel nach rechts.
- ^J (0Ah) Alpha-Cursor um 16 Pixel nach unten.
- ^K (0Bh) Alpha-Cursor um 16 Pixel nach oben.

- ^M (0Dh)** Alphamodus einschalten. Der Alpha-Cursor wird auf den Beginn der Pixelzeile gesetzt, die durch die letzte Grafik-Koordinate adressiert wurde.
- ^\_ (1Fh)** Alphamodus einschalten. Der Alpha-Cursor wird auf die zuletzt eingegebene Grafik-Koordinate gesetzt.
- ^Ö (1Ch)** Punktmodus einschalten.
- ^Ü (1Dh)** Vektormodus einschalten. Der nächste Vektor wird unterdrückt, da erst die Anfangskoordinate eingegeben werden muß. Darum kann dieser Befehl auch zum Eingeben einer Koordinate ohne Zeichnen verwendet werden.
- ^G (07h)** Sonderfunktion nach ^Ü: Der nächste Vektor wird doch gezeichnet. Sinnvoll, wenn der Vektormodus vorübergehend verlassen wurde und die Anfangskoordinate noch der letzten Koordinate entspricht.
- ^^ (1Eh)** Inkrementalmodus einschalten.
- ^P (10h)** Zeichne durch Ersetzen (REPLACE-Modus).
- ^Q (11h)** Zeichne mit Schwarz bzw. Hintergrundfarbe.
- ^R (12h)** Zeichne invers (Schwarz <-> Weiß, Vordergrundfarbe).
- ^S (13h)** Zeichne mit Weiß bzw. Vordergrundfarbe (default).

Im Grundmodus sind die letzten drei Befehle klar - löschen, invertieren und zeichnen. Im Farbmodus (s.u.) erfolgt Zeichnen und Löschen in den mit ESC 'V' bzw. ESC 'v' eingestellten Vorder- und Hintergrundfarben. Der Invers-Modus invertiert den 8-Bit-Farbindex.

Während bei diesen drei Befehlen nur die gesetzten Bits der Vektor-Maske (z.b. bei gestrichelten Vektoren) den Bildinhalt

verändern, werden im REPLACE-Modus die 1-Bits mit der Vordergrundfarbe (normalerweise weiß) und zugleich die 0-Bits der Maske mit der Hintergrundfarbe (schwarz) geschrieben. Der REPLACE-Modus wurde für GKS/GXS-Anpassungen implementiert.

## 2.2: Tektronix-Escapesequenzen

ESC 'A' a Y-Koordinatenursprung verschieben.

$$a = 20H + Y\text{-Koordinatenbereich}/16$$

Der Y-Koordinatenbereich erstreckt sich vom oberen Bildschirmrand bis zum Koordinatenursprung; defaultmäßig sind 560 Bildpunkte in Y-Richtung eingestellt, was einem a-Parameter 43h ('C') entspricht. <ESC> 'AD' setzt den Ursprung ganz nach unten, so daß die vollen 576 Pixelzeilen ausgenutzt werden.

ESC <^L> Löscht den Bildschirm, schaltet den Alpha-Modus ein und setzt den Cursor in die obere linke Ecke.

ESC 'G' x Zeichenattribute für den Alphamodus setzen. Die Attribute sind die gleichen wie im TVI-Modus (s.o.).

ESC ''' Linientyp: durchgehende Linie (default).

ESC 'a' Linientyp: gepunktete Linie.

ESC 'b' Linientyp: Strichpunkt-Linie.

ESC 'c' Linientyp: kurz gestrichelte Linie.

ESC 'd' Linientyp: lang gestrichelte Linie.

ESC 'P' Linientyp: Muster (Pattern).

Alle auf den Befehl <ESC>'P' folgenden Vektoren oder Flächen erhalten eine Musterstruktur, die der Darstellung des ASCII-Zeichens 7Fh entspricht und sich in jeder Richtung wiederholt. 7Fh ist defaultmäßig eine Pseudo-Graufäche mit Schachbrettmuster, kann aber wie jedes andere Zeichen über Zielkanal 4 (s.u.) undefiniert werden.

ESC 'S' a Linientyp: SPRAY.

Damit lassen sich 16 Schattierungswerte einstellen (a = '0'..'?'). Alle folgenden Vektoren oder Flächen werden mit einem Zufalls-Punktmuster geschrieben, wobei der Anteil der hell gesetzten Pixels dem Schattierungswert a entspricht (a = '0': völlig schwarz, a = '?': völlig weiß). Der SPRAY-Algorithmus verlangsamt die Vektor-Zeichengeschwindigkeit etwa auf die Hälfte.

ESC 'B' Blockmodus einschalten.

Im Blockmodus werden zwischen den Koordinaten anstelle von Vektoren Rechtecke gezeichnet, die in der aktuellen Farbe mit dem eingestellten Linientyp gefüllt sind. Die gesendeten Koordinaten bilden dabei die beiden diagonal gegenüberliegenden Ecken des Rechtecks.

Der Blockmodus eignet sich auch gut zum schnellen Löschen rechteckiger Bildbereiche. Er wird durch Einschalten des Vektor-Punkt-, Inkremental- oder Alphamodus wieder abgeschaltet.

ESC 'T' Zeichne gefülltes Dreieck.

Auf diesen Befehl müssen drei Koordinaten folgen; GRIP zeichnet dann ein in der aktuellen Farbe bzw. mit dem eingestellten Linientyp gefülltes Dreieck. Die Koordinaten bilden dabei die Ecken des Dreiecks.

Dieser Befehl läßt sich auch zum Zeichnen beliebiger gefüllter Polygone verwenden, da jedes Polygon aus Dreiecken zusammengesetzt werden kann.

### ESC 'F' Fülle Fläche.

Der Grafik-Cursor muß zum Flächenfüllen in einem vollständig umrandeten, konvexen Gebiet stehen. Die Umrandung wird in der mit <ESC> <ESC> 'v' (s.u.) eingestellten Bit-Ebene erkannt.

Das Füllen erfolgt in dem gewählten Linientyp (Maske, Muster oder Spray - s.o.). Ein beliebiges Polygon, das sich auch überschneiden darf, läßt sich mit <ESC> 'F' füllen, indem innerhalb jedes zweiten Eckpunktes das Flächenfüllen neu gestartet wird.

### 2.3: Allgemeines zum Vektormodus

Dieser Modus wird mit <^Ü> (1Dh) eingeschaltet. Als erstes muß danach eine Startkoordinate eingegeben werden. Das Bildfenster ist in ein X-Y-Raster von 768x576 Punkten eingeteilt. Im Interlaced-Betrieb entspricht jeder Koordinatenpunkt einem realen Bildschirmpunkt, während im Non-interlaced-Modus gerade und ungerade Y-Koordinaten (vertikal) auf den gleichen Bildpunkt abgebildet werden. Der Nullpunkt (0,0) liegt, im Gegensatz zum TVI-Modus, in der linken unteren Ecke. Es ist darauf zu achten, daß beim Zeichnen das Bildfenster nicht überschritten wird - aus Geschwindigkeitsgründen enthält GRIP keinen Clippingalgorithmus.

Das Zeichnen im Vektormodus geschieht durch die Eingabe von Koordinaten. Ein Vektor wird direkt nach der Eingabe von der letzten bis zu dieser Koordinate in der gewählten Farbe gezeichnet, es sei denn, der Koordinate ging unmittelbar ein <^Ü>-Befehl voraus. In diesem Fall wird sie nur als neue Startkoordinate übernommen, d.h. der nächste Vektor wird dann von da aus gezeichnet.

Eine vollständige Koordinate besteht aus 10 Bit in X-Richtung und 10 Bit in Y-Richtung. Zur Übermittlung wird der X- und der Y-Wert noch einmal in zwei Hälften zu je 5 Bits aufgeteilt. Die Koordinate wird also durch 4 ASCII-Zeichen übermittelt, bei denen jeweils die unteren 5 Bit (Bits 0-4) zum Koordinatenwert beitragen. Die oberen Bits (5-7) geben den Koordinatenteil (X oder Y, obere oder untere Hälfte) an.

*hy* = '...'?' (20h-3Fh): Obere 5 Bit der Y-Koordinate.  
*ly* = '""...' (60h-7Fh): Untere 5 Bit der Y-Koordinate.  
*hx* = '...'?' (20h-3Fh): Obere 5 Bit der X-Koordinate.  
*lx* = '@'...'\_' (40h-5Fh): Untere 5 Bit der X-Koordinate.

Beispiel: Die (X,Y)-Koordinate (500,200) schreibt sich binär (0111110100b, 0011001000b) und wird durch folgende Zeichen dargestellt:

*hy* = '&' (26h); *ly* = 'h' (68h); *hx* = '/' (2Fh); *lx* = 'T' (54h).

Die Übertragung erfolgt in der Reihenfolge *hy,ly,hx,lx*. In TURBO-PASCAL sieht dann eine Routine zur Koordinatenübertragung bzw. zum Vektorzeichnen etwa so aus:

```
procedure PLOTXY (X,Y: integer);
(* Zeichne Linie zum Punkt (X,Y). *)
begin
  write (chr(Y div 32 + 32),chr(Y mod 32 + 96),
        chr(X div 32 + 32),chr(X mod 32 + 64))
end;
```

Es läßt sich eindeutig unterscheiden, zu welchem Koordinatenteil ein Zeichen gehört (bis auf *hx* und *hy*, die identisch kodiert werden). Dies kann ausgenutzt werden, wenn eine Koordinate sich nur teilweise ändert. GRIP speichert die zuletzt eingegebenen Werte und

faßt jede Teilfolge, die mit *lx* abgeschlossen wird, als neue Koordinate auf. Die gleichbleibenden Teile brauchen dabei nicht erneut eingegeben zu werden.

Die Koordinate bleibt auch dann gespeichert, wenn der Vektormodus zwischenzeitlich verlassen wird. Es sind folgende Formate zur Übertragung einer Koordinate erlaubt:

*hy ly hx lx*: Koordinate völlig neu.  
*hy lx*: *ly* und *hx* beibehalten.  
*ly hx lx*: *hy* beibehalten.  
*ly lx*: *hy* und *hx* beibehalten.  
*lx*: *Y* und *hx* beibehalten.

Die Koordinate (500,200) aus dem obigen Beispiel kann also als '&h/T' oder - wenn sie Teile mit der vorherigen Koordinate gemeinhat - als '&T', 'h/T', 'hT' oder einfach als 'T' gesendet werden.

Die obige TURBO-PASCAL-Routine mit abgekürzter Koordinatenübertragung sieht schon etwas länger aus:

```

procedure PLOTXY (X,Y:integer);
var NXH,NXL,NYH,NYL: char;
(* XH,YH,XL,YL: char, global definiert *)
begin
  NYH:=chr(Y div 32 + 32);NYL:=chr(Y mod 32 + 96);
  NXH:=chr(X div 32 + 32);NXL:=chr(X mod 32 + 64);
  if YH=NYH then
    if YL=NYL then
      if XH=NXH then write(NXL)
        else write(NYL+NXH+NXL)
      else if XH=NXH then write(NYL+NXL)
        else write(NYL+NXH+NXL)
      else if (YL=NYL) and (XH=NXH)
        then write(NYH+NXL)
          else write(NYH+NYL+NXH+NXL);

```

```
XH:=NXH;YH:=NYH;XL:=NXL;YL:=NYL;  
end;
```

Nachdem die erste Koordinate im Vektormodus übertragen wurde, zeichnet GRIP von dort zur nächsten und von dieser zu jeder weiteren Koordinate eine Linie, so daß auf dem Bildschirm nach und nach ein Polygonzug entsteht. Der Empfang des *lx*-Bytes gibt dabei immer das Signal zum Zeichnen des nächsten Vektors. Der Linientyp kann mit den Sequenzen <ESC> ''' - <ESC> 'd' etc. bestimmt werden, die Farbe mit <ESC> 'V', die Blinkfarbe mit <ESC> 'Y' (s.u.).

Zwei aufeinanderfolgende gleiche Koordinaten erzeugen nur einen Punkt. Mit <^Ü> (1Dh) läßt sich die Linie zur nächsten Koordinate unterdrücken, erst der übernächste Vektor wird wieder gezeichnet. Die Linienunterdrückung kann jedoch mit <^G> (07h) wieder aufgehoben werden. Dies ist sinnvoll, wenn man den Vektormodus kurz verläßt und dann mit <^Ü> wieder zurückkehrt. Nach Eingabe von <^G> kann dann gleich weitergezeichnet werden, während man sonst die Startkoordinate neu eingeben müßte.

Die Vektoren werden normalerweise in der Vordergrundfarbe (weiß) gezeichnet. Da das klassische Tektronix-Terminal eine Speicherbildröhre besitzt, kann es einmal gezeichnete Linien nicht mehr einzeln löschen. Bei GRIP ist dies jedoch kein Problem. Deshalb wurden noch einige Befehle implementiert (<^P>,<^Q>,<^R>,<^S>), die im ursprünglichen Tektronix-Befehlssatz nicht enthalten sind.

Das Löschen sollte in der gleichen Richtung wie das Zeichnen erfolgen, sonst können einige Teile des Vektors stehen bleiben. Die Übergänge zwischen den Pixelzeilen werden nach einem Algorithmus berechnet, der rückwärts die "Stufen" anderswohin setzt als vorwärts.

Zu beachten ist, daß das Zeichnen senkrechter und vor allem waagerechter Linien um ein vielfaches schneller erfolgt als das Zeichnen schräger Linien.

## 2.4: Punktmodus

Der Punktmodus funktioniert genauso wie der Vektormodus, allerdings wird keine Linie, sondern nur der Endpunkt gezeichnet. Die Lösch- und Invertierungsbefehle wirken auch hier.

## 2.5: Inkrementalmodus

Im Inkrementalmodus wird von der letzten Grafikkordinate an in 1-Punkt-Schritten weitergezeichnet; die gespeicherte Koordinate läuft entsprechend mit. Das Ganze funktioniert ähnlich wie ein mechanischer Plotter mit Schrittmotoren. Die Richtung wird durch die folgenden Zeichen vorgegeben:

			'D' (44h): nach oben.
'F'	'D'	'E'	'E' (45h): diagonal nach rechts oben.
			'A' (41h): nach rechts.
			'T' (49h): diagonal nach rechts unten.
'B'	*	'A'	'H' (48h): nach unten.
			'J' (4Ah): diagonal nach links unten.
			'B' (42h): nach links.
'J'	'H'	'T'	'F' (46h): diagonal nach links oben.

Die Zeichenmodi werden über ' ', 'P', 'Q', 'R', 'T' eingestellt. Jeder Modus bleibt so lange erhalten, bis er von dem nächsten abgelöst wird. Direkt nach Einschalten des Inkrementalmodus muß einer dieser Befehle gegeben werden:

' ' (20h): Punkte nicht zeichnen (nur Grafik-Cursor bewegen).

'P' (50h): Punkte zeichnen (Weiß, wie ^S).

'Q' (51h): Punkte löschen (Schwarz, wie ^Q).

'R' (52h): Punkte invertieren (wie ^R).

## 3

Farben

GRIP-6B unterstützt von sich aus acht Farben bzw. Helligkeitsstufen für Text und Grafik, GRIP-6C sogar 128 Farben - allerdings nur im Textmodus. Die Farbgrafik ist 'zweistufig': In den Bildspeicher wird nur ein Farbindex (7 Bit bei GRIP-7 bzw. 3 Bit bei GRIP-6) geschrieben. Die Zuordnung zwischen Farbindex und Farbton erfolgt über eine Look-Up-tafel.

3.1 Farbindizes

Farbindizes lassen sich über die Textattribut-Befehle im VT100-Modus einstellen. Allgemeingültiger sind die folgenden Befehle, die in allen Modi funktionieren:

ESC 'V' x	Vordergrundfarbe einstellen.
ESC 'v' x	Hintergrundfarbe einstellen.

In dem gewählten Farbindex x (00h..FFh) werden von nun an alle Grafiken und Buchstaben geschrieben. Bei GRIP-6 sind nur die letzten 3 Bits des Farbindex relevant.

Der Vordergrund-Farbindex gilt im Tektronix-Modus für das Zeichen (^S), im Text-Modus wird darin das Textzeichen geschrieben. Der Hintergrund-Farbindex gilt im Tektronix-Modus für das Löschen (^Q) und im Text-Modus für den Zeichenhintergrund. Defaultmäßig ist Farbindex 47h (GRIP-6C) bzw. 07h (GRIP-6B) für den Vordergrund und Farbindex 00h für den Hintergrund eingestellt.

### 3.2 Farbdarstellung und Look-Up-Tafel

Die Farbe, in der ein bestimmter Farbindex am Bildschirm sichtbar ist, läßt sich über die Look-Up-Tafel mit den folgenden Befehlen einstellen:

ESC ESC 'W' x r g b	Farbton zuordnen.
ESC ESC 'Y' x r g b	Blink-Farbton zuordnen.

x = Farbindex (0..255)

r = Rot-Intensität (0..63) + 30h

g = Grün-Intensität (0..63) + 30h

b = Blau-Intensität (0..63) + 30h

Der Befehl programmiert die Look-Up-Tafel für den Farbindex x um. Bei GRIP-6C kann Farbton, Farbsättigung und Helligkeit unabhängig über die Rot-, Grün- und Blau-Intensität in je 64 Stufen bestimmt werden. '0' stellt z.b. die niedrigste, 'P' eine mittlere, 'o' die höchste Intensität ein. Es ergeben sich durch die Mischung der drei Farbanteile insgesamt 262.144 einstellbare Farbtöne.

Da sich die Farbtöne auch nachträglich umdefinieren lassen, kann z.b. durch kontinuierliche Veränderung der Helligkeit ein Bild 'aufgeblendet' werden.

Bei GRIP-6B sind nur die letzten drei Bits des Farbindex und nur das letzte Bit der Intensitätswerte relevant, d.h. die Parameter r,g,b können nur die Werte '0' und '1' annehmen. Auch der Grauwert im BAS-Signal wird über die Look-Up-Tafel eingestellt. Dies geschieht nach folgendem Schema:

r g b	Farbe	Intensität
0 0 0	Schwarz	0 (dunkel)
0 0 1	Blau	1
0 1 0	Grün	2
0 1 1	Cyan	3
1 0 0	Rot	4
1 0 1	Purpur	5
1 1 0	Gelb	6
1 1 1	Weiß	7 (hell)

Die Bildränder (Austastlücke) haben den Farbindex 00h. **Zu beachten** ist, daß einige Farbmonitore zur korrekten Farbdarstellung eine schwarze Hintergrundfarbe benötigen, d.h. der Farbindex 00 muß für diese Monitore immer als schwarz definiert sein.

Um einen Farbindex zwischen zwei Farbtönen blinken zu lassen, wird mit <ESC><ESC>'Y' dem Farbindex x ein Blink-Farbton zugeordnet. Dieser Befehl muß unmittelbar nach Zuordnen des normalen Farbtons (<ESC><ESC>'W') gesendet werden. <ESC><ESC>'W' allein schaltet das Blinken für den entsprechenden Farbindex ab.

Beispiel: <ESC> <ESC> 'W' 07h 'o00' <ESC> <ESC> 'Y' 07h '0o0' läßt den Farbindex 07h zwischen Rot und Grün blinken.

### 3.3 Sonderbefehle

Zum Auslesen einer Ebene des Bildspeichers (s.u.) läßt sich die Bitnummer des Grund- bzw. Farb-RAM mit folgendem Befehl einstellen:

ESC ESC 'v' a                      Bildspeicher-Ebene einstellen.

a = '0': Grundebene (def)	'@': Grundebene, invertiert
'1': Farbebene 0	'A': Farbebene 0, invertiert
'2': Farbebene 1	'B': Farbebene 1, invertiert
'3': Farbebene 2	'C': Farbebene 2, invertiert
'4': Farbebene 3	'D': Farbebene 3, invertiert
'5': Farbebene 4	'E': Farbebene 4, invertiert
'6': Farbebene 5	'F': Farbebene 5, invertiert
'7': Farbebene 6	'G': Farbebene 6, invertiert
'8': Farbebene 7	'H': Farbebene 7, invertiert
'9': Summenwert	

Dieser Befehl ist notwendig für Hardcopy, Flächenfüllen oder zum Auslesen des Video-RAM's über Quellenkanal 3 (s.u.). Die Farbebene 0 besteht aus den Bits 0 der Farbindices; entsprechendes gilt für die anderen Ebenen. Bei GRIP-6B existieren nur die Ebenen 0-2, bei GRIP-6A nur die Ebene 0. Der Parameter '9' liefert einen Summenwert, der sich aus einer OR-Funktion über die drei Farbebenen ergibt.

#### 4 Die Tastatur

Normalerweise werden Typ und Parameter der Tastatur über das Set-Up-Menü ausgewählt. In Sonderfällen können dazu auch die folgenden Befehle benutzt werden:

ESC ESC '4' x     Definiere Tastaturkanal und -Parameter.

x = 0 w 0 c1-0 h k1-0

w = 0: Tastatur-Wort 8 Bit     h = 0: Host-Wortlänge 8 Bit  
     1: Tastatur-Wort 7 Bit     1: Host-Wortlänge 7 Bit (def)

k1-0 = 00: Tastenklick aus (def)  
     01: leiser Tastenklick  
     10: normaler Tastenklick  
     11: lauter Tastenklick

c1-0 = 01: PC/AT-Tastatur  
     10: Serielle Tastatur (V24 oder TTL)  
     11: Tastatur abgeschaltet.

ESC ESC '3' a     Spezifiziere serielle Tastatur-Baudrate.

a = '0': Baudrate wie J7 (def)  
     '1': 50 Baud     '7': 1200 Baud  
     '2': 75 Baud     '8': 2400 Baud  
     '3': 110 Baud    '9': 4800 Baud  
     '4': 150 Baud  
     '5': 300 Baud  
     '6': 600 Baud

Der Default-Tastaturkanal wird über Jumper J7 (-> 'Installation') bzw. über das Set-Up-Menü eingestellt. Jeder Tastaturkanal läuft über einen 64 Byte FIFO-Zwischenspeicher (Tastaturpuffer). Wenn auf

einem der Tastaturkanäle ein Tastencode gesendet wird, gelangt er zunächst in den zugeordneten Tastaturpuffer. Ist der Tastaturkanal aktiv, d.h. mit den Bits c1-0 ausgewählt, so werden die Tastencodes aus dem Puffer nacheinander über die -> Umcodetabelle umcodiert (falls dort ein Eintrag für diesen Tastencode steht) und dann zum Host übertragen.

Der 'Tastenklick' ist ein kurzer Knackton, der bei Betätigung einer Taste zur akustischen Rückmeldung ertönt.

Das w-Bit bezieht sich auf das Byte von der Tastatur, das h-Bit auf das Byte, das über den Tastaturkanal zum Host gesendet wird. Mit w=0 und h=1 kann das 8. Bit der Tastatur ausgeblendet, aber trotzdem noch in der Umcodetabelle ausgewertet werden, z.b. bei Funktionstasten, die das 8. Bit auf 1 setzen. Normalerweise sollte das h-Bit immer auf 1 gesetzt sein, um am 8. Bit Tastaturcodes (8. Bit = 0) von sonstigen Daten (8. Bit = 1) zu unterscheiden (-> 'Quellenkanäle').

Das Abschalten der Tastatur (c1-0 auf 11) ist sinnvoll, wenn vom Quellenkanal (s.u.) ein längerer Datenstrom erwartet wird, z.b. beim direkten Auslesen des Video-RAM's, der nicht von Tastaturcodes unterbrochen werden soll.

Beispiel: <ESC><ESC>'46' (1Bh 1Bh 34h 36h) paßt GRIP an eine serielle Tastatur an und setzt die Tastenklick-Lautstärke auf "normal".

## 5: Ziel- und Quellenkanäle

Unter diesem Begriff sind Funktionsgruppen des GRIP-Terminals zusammengefaßt, an die der Hostrechner Daten senden kann (Zielkanäle), wie z.B. das Druckerinterface oder eine der internen Tabellen. Ebenso gibt es Kanäle, von denen der Hostrechner Daten empfangen kann (Quellenkanäle), etwa die Tastatur oder das Video-RAM.

Zwei der Kanäle wurden bereits beschrieben, die Text-Konsole (TVI- oder VT100-Modus) und die Grafik-Konsole (Tektronix-Modus). Vom Host gesendete Zeichen gehen normalerweise zu einer der Konsolen, um dort als Buchstaben oder Grafik auf dem Bildschirm zu erscheinen. Wenn der Host von GRIP Daten empfängt, kommen diese, wie für ein Terminal zu erwarten, im allgemeinen von der Tastatur.

Neben den Konsolen und dem Tastaturkanal kennt GRIP-6 noch zwölf weitere Zielkanäle und fünfzehn Quellenkanäle für Daten. Die meisten Kanäle übertragen in beide Richtungen, sind also gleichzeitig Ziel- und Quellenkanal.

Wenn zwischen GRIP und Host 8-Bit-Übertragung möglich ist, wie z.B. beim ECB-Bus, dann entscheidet beim Datentransfer Host->GRIP das achte Datenbit über das Datenziel. Ist es 0 (wie bei den normalen ASCII-Zeichen 20h..7Fh), geht das Byte zu einer der Konsolen; ist es 1, wird es zu einem eingestellten Zielkanal gesendet. Dabei wird das 8. Bit vor der Übertragung auf 0 maskiert, so daß wieder ein ASCII-Zeichen herauskommt.

Der Empfang von Daten funktioniert ähnlich: Daten mit gesetztem 8. Bit kommen im allgemeinen vom Quellenkanal und nicht von der Tastatur (Voraussetzung: h-Bit=1, s.o.). Der Datenempfang von einer anderen Quelle als der Tastatur muß mit einem Abfrage-Befehl eingeleitet werden. Einen solchen Befehl haben wir schon kennengelernt: <ESC> '?' fragt im TVI-Modus die Cursorposition ab.

Wenn auch von der Tastatur 8 Bits kommen können (h-Bit=0), oder wenn nur eine 7-Bit-Verbindung zum Host besteht, oder wenn beim Zielkanal das achte Bit irgendeine Bedeutung hat und nicht einfach ausgeblendet werden darf, ist die Unterscheidung nicht so einfach. In diesen Fällen müssen die Daten durch spezielle Transferbefehle (s.u.) 'umgeleitet' werden.

## 5.1 Kanalwahl und Transfer

Folgende Dopplescape-Sequenzen wählen einen der Kanäle aus:

ESC ESC 'C' a Zielkanal wählen.  
 ESC ESC 'c' a Quellenkanal wählen.

a =	'0': Centronics	'8': XR-Bildinhalt
	'1': ECB-Bus	'9': User-Statuszeile
	'2': RS232-Schnittstelle	'A': Systempatch
	'3': Video-RAM	'B': Druckerpatch
	'4': Zeichensätze	'C': -
	'5': Userprogramm	'D': Serielle Tastatur
	'6': Tasten- Umcodetabelle	'E': PC/AT-Tastatur
	'7': Tongenerator	

Der Default-Quellenkanal ist die RS232-Schnittstelle. Kanal 0 ist als Quellenkanal inaktiv. Die Tastaturschnittstellen können nicht als Zielkanal und nur dann als Quellenkanal benutzt werden, wenn sie nicht bereits als Terminal-Tastaturkanal definiert sind. Ziel- oder Quellenkanal dürfen mit dem Host-Kanal identisch sein; damit kann man z.B. während einer Hardcopy (s.u.) die Druckerdaten zum Host umleiten.

Die folgenden Befehle führen den Datentransfer durch:

ESC ESC 'D' aa    Sende ein Byte im ASCII-Hex-Format zum Zielkanal.

ESC ESC 'd' x     Sende ein Byte binär zum Zielkanal.

ESC ESC 'F' aa bb..<CR>    Byte-Folge im ASCII-Hex-Format zum Zielkanal senden.

Beendet wird die Folge mit <CR> (0Dh). Die einzelnen Bytes können - müssen aber nicht - durch Spaces (20h) getrennt werden.

ESC ESC 'f' xx y z..    Byte-String zum Zielkanal senden.

xx gibt die Anzahl der Bytes an (1-65535, Reihenfolge LSB,MSB).

80h-FFh            Direktmodus (8.Bit gesetzt): Byte mit ausmaskiertem 8. Bit direkt zum Zielkanal senden.

Dieser Modus ist immer eingestellt und stellt die einfachste Art der Übertragung zum Zielkanal dar.

Beispiele: Um die ASCII-Zeichenfolge '+123'<CR> zum Zielkanal zu senden, gibt es folgende Möglichkeiten:

Direktmodus: ABh B1h B2h B3h 8Dh.

ASCII/Hex: <ESC> <ESC> 'F2B3132330D' <CR>.

String: <ESC> <ESC> 'f 05h 00h '+123' <CR>.

Ähnliche Befehle existieren zum Empfang vom Quellenkanal:

ESC ESC 'E' /aa    Hole ein Byte im ASCII-Hex-Format vom Quellenkanal.

Das 8. Bit der beiden empfangenen ASCII-Zeichen ist gesetzt. Sind vom Kanal keine Daten verfügbar (z.B. Empfangspuffer leer bei RS232), so wird stattdessen ein <CR>-Zeichen mit gesetztem 8. Bit (8Dh) gesendet.

ESC ESC 'e' /x     Hole ein Byte vom Quellenkanal.

Sind keine Daten verfügbar, so wird stattdessen eine Null (00h) gesendet.

ESC ESC 'G' aaaa /bb..<CR>     Mehrere Bytes im ASCII-Hex-Format mit gesetztem 8. Bit vom Quellenkanal holen.

Die Hex-Zahl aaaa gibt die Maximal-Anzahl der zu empfangenden Bytes an (0001-FFFF). Sobald aaaa Bytes gesendet wurden oder keine Daten mehr verfügbar sind, wird die Übertragung mit <CR> mit gesetztem 8. Bit (8Dh) beendet.

ESC ESC 'g' xx/y z..     Mehrere Bytes vom Quellenkanal holen.

Das Wort xx gibt die Anzahl an und kann den Wert von 1-65535 (FFh FFh) annehmen. Sind keine Daten mehr verfügbar, werden Nullen (00h) gesendet, bis die Anzahl xx erreicht ist.

Die Datenübertragung zu Schnittstellen oder internen Tabellen erfolgt also in zwei Schritten: Zunächst muß der Zielkanal mit <ESC><ESC>'C' spezifiziert werden, danach erfolgt die eigentliche Übertragung entweder durch gesetztes 8. Bit oder durch einen Transferbefehl. Der einmal gewählte Kanal bleibt bis zur nächsten Änderung eingestellt.

Das Lesen eines Bytes vom Quellenkanal, z.b. von der RS232-Schnittstelle, erfordert insgesamt folgende Schritte:

1. Quellenkanal wählen, z.b. <ESC><ESC>'c2' für die RS232-Schnittstelle.
2. Hex-Abfragebefehl senden, z.b. <ESC><ESC>'E'.

3. GRIP abfragen, bis ein Byte das 8. Bit gesetzt hat. Voraussetzung: h-Bit des Tastaturkanals (s.o.) steht auf 1!
4. Überprüfen, ob das gelesene Byte <CR> (8Dh) ist. Wenn ja, gibt es keine Daten mehr, und man ist fertig. Wenn nein, zweites Byte holen.
5. Bei beiden Bytes das 8. Bit auf 0 maskieren, Hex->Binär-Konvertierung durchführen, das so gewonnene Zeichen abspeichern, ein neues Byte von GRIP holen und zu Schritt 3 zurückgehen.

Im Kapitel 'RS232-Schnittstelle' (s.u.) ist das Ganze in Z80-Assemblersprache realisiert. Die einzelnen Kanäle werden in den nächsten Abschnitten näher erläutert.

## 5.2 Parallelport/Centronics-Schnittstelle (Zielkanal 0)

An diese Schnittstelle wird normalerweise ein Drucker angeschlossen. Die Datenübertragung sollte der Einfachheit halber im Direktmodus erfolgen. Es ist auch möglich, die Centronics-Schnittstelle als 8-Bit-Parallelport für Steuerzwecke zu benutzen.

Der Centronics-Schnittstelle kann ein 32-Kbyte-Spooler vorgeschaltet werden (s.u.), der etwa 15 Seiten Text faßt. Es folgt als Beispiel eine Implementierung als LST:-Device im CP/M-BIOS (GRIPOUT, GRPOST siehe Kapitel 'Installation'):

CENOUT sendet das Zeichen in Register C an die Centronics-Schnittstelle.

```
CENOUT: SET 7,C      ;Direktmodus: Achstes Bit auf 1
          JP  GRIPOUT ;Ausgabe an Zielkanal
```

Alternative CENOUT-Routine, wenn Direktmodus nicht möglich ist  
(z.b. nur 7-Bit-Übertragung von Host):

```

CENOUT:  PUSH      BC           ;Zeichen retten
          LD        C,1BH       ;Übertragungsbefehl geben
          CALL     GRIPOUT      ;sende <ESC>
          CALL     GRIPOUT      ;sende <ESC>
          LD        C,64H       ;'d'
          CALL     GRIPOUT      ;
          POP      BC           ;Zeichen zurück
          JP       GRIPOUT      ;1 Byte direkt zum Zielkanal

```

CENST prüft den Status des LST:-Device.

```

CENST:   JP       GRIPOST      ;Solange GRIP ready ist, ist auch
                                   ;der Drucker-Kanal ready.

```

Der ERROR-Eingang der Centronics-Schnittstelle, der normalerweise nicht benutzt wird, kann über den Lichtgriffel-Befehl (s.u.) abgefragt werden.

### 5.3 Die ECB-Bus-Schnittstelle (Ziel/Quellenkanal 1)

Hier wird normalerweise der Hostrechner angeschlossen. Für Sonderzwecke, z.B. zur Kopplung zweier Rechner, kann es jedoch sinnvoll sein, die ECB-Bus-Schnittstelle als normalen Übertragungskanal zu einem zweiten Rechner zu benutzen. Die Schnittstelle ist als Quellenkanal mit 256 Byte FIFO-Speicher gepuffert. Installation siehe Hardware-Handbuch.

### 5.4 Die serielle Schnittstelle (Ziel/Quellenkanal 2)

Wenn die RS232- oder RS422-Schnittstelle nicht vom Host belegt wird, kann sie zum Anschluß von Peripheriegeräten (Drucker, Maus) implementiert werden. Sender und Empfänger lassen sich mit verschiedenen Geschwindigkeiten betreiben. Installation siehe Hardware-Handbuch.

Die Parameter lassen sich sowohl im Host- wie auch im Schnittstellenbetrieb mit folgenden Befehlen ändern:

ESC ESC '0' a Spezifiziere Empfängerbaudrate.

ESC ESC '1' a Spezifiziere Senderbaudrate.

a =	'0': Baudrate aus EEPROM (default)	
	'1': 50 Baud	'7': 1200 Baud
	'2': 75 Baud	'8': 2400 Baud
	'3': 110 Baud	'9': 4800 Baud
	'4': 150 Baud	':': 9600 Baud
	'5': 300 Baud	':': 19200 Baud
	'6': 600 Baud	'<': 38400 Baud

Beispiel: <ESC> <ESC> '17' (1Bh 1Bh 30h 37h) setzt die Senderbaudrate auf 1200 Baud.



Centronics-Schnittstelle, der Spooler zuschalten. Das ist sinnvoll, wenn z.B. ein Drucker über diese Schnittstelle implementiert wird.

Die folgenden Routinen im Z80-Assemblercode steuern die Schnittstelle als AUX:-Device an. Zunächst eine Hilfsroutine für Doppelescape-Sequenzen:

DESCOUT sendet zwei <ESC>-Zeichen an GRIP, gefolgt von dem Zeichen in Register C.

```
DESCOUT: PUSH BC           ;Befehlszeichen retten
          LD  C,1BH        ;<ESC>
          CALL GRIPOUT     ;Ausgabe; siehe 'Installation'
          CALL GRIPOUT     ;Zweimal ESC
          POP  BC          ;Befehlszeichen zurück
          JP   GRIPOUT     ;Zeichen ausgeben
```

SEROUT sendet das Zeichen in Register C an die RS232-Schnittstelle.

```
SEROUT:  PUSH BC           ;Zeichen retten
          LD  C,43H        ;'C' für Zielkanal
          CALL DESCOUT     ;<ESC> <ESC> 'C' ausgeben
          LD  C,32H        ;'2' für RS232
          CALL GRIPOUT     ;Zielkanal auf RS232
          POP  BC          ;Zeichen zurück
          SET  7,C         ;Achstes Bit setzen für Direktmodus
          CALL GRIPOUT     ;Zeichen senden an RS232
          LD  C,43H        ;'C' für Zielkanal
          CALL DESCOUT     ;<ESC> <ESC> 'C' ausgeben
          LD  C,30H        ;'0' für Centronics
          JP   GRIPOUT     ;Zielkanal zurück auf Centronics
```

Die zugehörige Statusroutine SEROST für den Senderstatus ist wegen der internen FIFO-Pufferung in der GRIP wieder sehr einfach:

SEROST: JP GRIPOST ;siehe Installation

Anders verhält es sich mit SERIST, die den Empfängerstatus abfragt (Register A = FFH wenn ready, 00H wenn nicht ready):

```

SERIST:  LD  C,34H      ;'4' für Tastaturparameter
         CALL DESCOUT  ;<ESC> <ESC> '4' ausgeben
         LD  C,1CH      ;Byte für "Tastatur aus"
         CALL GRIPOUT  ;Tastatur abschalten
         LD  C,63H      ;'c' für Quellenkanal
         CALL DESCOUT  ;<ESC> <ESC> 'c' ausgeben
         LD  C,32H      ;'2' für RS232
         CALL GRIPOUT  ;Quellenkanal auf RS232
         LD  C,45H      ;'E' für ASCII-Hex-Transfer
         CALL DESCOUT  ;<ESC> <ESC> 'E' ausgeben

SERTST:  CALL GRIPIN   ;GRIP abfragen
         BIT  7,A       ;Byte noch von Tastatur?
         JR   Z,SERTST  ;Falls ja, warten

         AND  7FH       ;8. Bit maskieren
         LD  (NIB1),A   ;Erstes Nibble abspeichern
         CP  0DH       ;<CR> -> keine Daten im FIFO?
         JR   Z,ENDE    ;Falls ja

         CALL GRIPIN   ;Zweites Nibble holen
         AND  7FH       ;8. Bit maskieren
         LD  (NIB2),A   ;Zweites Nibble abspeichern
         CALL GRIPIN   ;Muß jetzt <CR> sein

ENDE:    LD  C,34H      ;'4' für Tastaturparameter
         CALL DESCOUT  ;<ESC> <ESC> '4' ausgeben
         LD  C,TASTDEF  ;Byte für Tastaturparameter
         CALL GRIPOUT  ;Tastatur wieder einschalten

```

```

LD   A,(NIB1)   ;Erstes empfangenes Byte prüfen
CP   0DH        ;<CR> -> keine Daten verfügbar?
LD   A,00H      ;Not-ready-Zeichen
RET  Z          ;
Ld   A,FFh      ;FFh = Ready-Zeichen
RET

```

```

NIB2   DS   1      ;Speicherplatz für
NIB1   DS   1      ;RS232-Daten (ASCII-Hex)

```

SERIN holt ein Zeichen von der RS232-Schnittstelle nach Register A:

```

SERIN: LD   A,(NIB1)   ;Erstes Nibble bei der letzten Abfrage
        CP   0DH        ;<CR> -> keine Daten verfügbar?
        CALL Z,SERIST  ;Falls ja, nochmal abfragen
        JR   Z,SERIN   ;Warten bis ready

        LD   HL,(NIB2) ;H=Erstes, L=Zweites Nibble
        JP   CONVERT  ;ASCII-Hex (HL) -> Binär (A) kon-
                        ;vertieren

```

### V.5.5 Video-RAM-Transfer (Ziel/Quellenkanäle 3, 8 und 12)

Dieser Kanal dient zum direkten Zugriff vom Host auf den Bildschirmspeicher. Er erlaubt nicht nur schnelle Bildtransfers, sondern auch z.B. die Benutzung eines nicht benötigten Speicherbereichs als RAM-Floppy. Wenn im Non-Interlaced-Modus Spooler und zweite Bildschirmseite abgeschaltet sind (-> <ESC><ESC> '5'), steht die Hälfte des Video-RAM's (inkl. XR-RAM) für solche Zwecke zur Verfügung.

Der Bildschirmspeicher von GRIP-6 ist folgendermaßen organisiert: Jeweils acht aufeinanderfolgende Bytes bilden auf dem Schirm einen rechteckigen Block von 64 (8x8) Pixels. Jeder Block kann ein ganzes Textzeichen im Non-Interlaced-Modus oder ein halbes im Interlaced-Modus enthalten. Diese spezielle Bildspeicher-Organisation beschleunigt den Textaufbau, da ein Zeichen mit einem zusammenhängenden DMA-Transfer übertragen werden kann. Die einzelnen Bytes bilden von oben nach unten die Pixelzeilen des Blocks. Ein Ausschnitt des Bildschirms, beginnend an einer Blockgrenze mit Byte N, ist im RAM nach folgendem Schema angeordnet:

```

Bit   0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 . .
Byte  N x x x x x x x N+8 x x x x x x N+16 x x x x x N+24 . .
. . . N+1 x x x x x x N+9 x x x x x x N+17 x x x x x N+25 . .
. . . N+2 x x x x x x N+10 x x x x x N+18 x x x x x N+26 . .
. . . N+3 x x x x x x N+11 x x x x x N+19 x x x x x N+27 . .
. . . N+4 x x x x x x N+12 x x x x x N+20 x x x x x N+28 . .
. . . N+5 x x x x x x N+13 x x x x x N+21 x x x x x N+29 . .
. . . N+6 x x x x x x N+14 x x x x x N+22 x x x x x N+30 . .
. . . N+7 x x x x x x N+15 x x x x x N+23 x x x x x N+31 . .
. . . N+768 x x x x x N+776 x x x x x N+784 x x x x x N+792 . .
. . . N+769 x x x x x N+777 x x x x x N+785 x x x x x N+793 . .
. . .
. . .
. . .
. . .

```

Bei Kanal 3 (Monochrom-Übertragung) entspricht ein Byte acht nebeneinanderliegenden Pixeln. Die Pixel werden aus der mit <ESC><ESC> 'v' (s.o.) eingestellten Speicherebene ausgelesen (1=hell,

0=dunkel) bzw. mit der eingestellten Vordergrundfarbe parallel in die selektierten Ebenen eingeschrieben.

Die angesprochene Adresse im Video-RAM wird mit folgenden Befehlen eingestellt:

ESC ESC 'T' aaaa Setze Video-RAM-Adresse im ASCII-Hex-Format.

ESC ESC 'i' xx Setze Video-RAM-Adresse (binär).

Die eingestellte Adresse wird alle 8 Pixel hochgezählt. 0000h entspricht der oberen linken Ecke. Die Speicherebene wird mit <ESC><ESC>'v' (s.o.) eingestellt. Bit 15 der Adresse aaaa legt im Non-Interlaced-Betrieb fest, in welche Bildschirmseite geschrieben wird (-> ESC ESC '5'):

Bit 15 = 0 : Übertragen in Vordergrundseite

1 : Übertragen in Hintergrundseite

Wenn im Non-Interlaced-Modus weder Spooler noch Hintergrundseite benötigt werden, kann der freie 32-KB-Bereich z.b. als RAM-Floppy implementiert werden, auf die man dann mit Bit 15 = 1 zugreifen kann.

## 5.6 Die Zeichensätze (Ziel/Quellenkanal 4)

GRIP's akkugepuffertes RAM bietet Platz für bis zu 18 verschiedene Zeichensätze, je nach Auflösung. Diese Zeichensatz-Bereiche können zum einen aus dem EPROM mit vordefinierten Zeichen geladen werden, zum anderem kann der Host über Kanal 4 frei darauf zugreifen.

Die 18 vordefinierte Zeichensätze beinhalten acht ASCII-Zeichensätze, einen Spezialzeichensatz mit dem griechischen Alphabet und mathematischen Sondersymbolen, und verkleinerte Versionen all dieser Zeichensätze für die Indexschrift.

Zwischen Standard- und Indexzeichensatz kann man im TVI-Modus mit <ESC> 'G' (s.o.) umschalten. Im VT-100-Modus steht der Befehl <ESC> '' zur Zeichensatzwahl zur Verfügung. Alles andere erledigt der folgende Befehl:

ESC ESC '7' a      Zeichensatz auswählen oder laden.

a =	'0'	Zeichensatz USA (def)	'8'	Zeichensatz Griechisch
	'1'	Zeichensatz Britisch	'9'	Lade alle Zeichensätze
	'2'	Zeichensatz Deutsch	':'	Lade ASCII- Standard
	'3'	Zeichensatz Dänisch	';'	Lade ASCII- Index
	'4'	Zeichensatz Französisch	'<'	Lade Griechisch-Standard
	'5'	Zeichensatz Schwedisch	'='	Lade Griechisch-Index
	'6'	Zeichensatz Italienisch		
	'7'	Zeichensatz Spanisch		

Die vier aktuellen Zeichensätze stehen resident im RAM und können über den Kanal 4 (s.u.) umdefiniert werden (s.u.). Beim Umschalten der acht ASCII-Zeichensätze (Parameter '0'-'7') werden lediglich die nationalen Sonderzeichen ausgetauscht. Der Parameter '9' kopiert alle vier Zeichensätze aus dem EPROM ins RAM, ':' - '=' jeweils nur einen.

Alle Zeichen lassen sich gleichzeitig auf dem Schirm darstellen; die Umschaltung auf einen neuen Zeichensatz wirkt auch mitten in einer Zeile. Sie gilt nur für die folgenden Zeichen, die schon geschrieben verändern sich nicht.

Bei einer Zeichenmatrix größer 8x8 (z.b. im Interlaced-Modus) passen nur halb so viele Zeichen in das RAM. Der Befehl <ESC> <ESC> '78' überschreibt darum dort die ASCII-Zeichensätze mit den griechischen Zeichensätzen.

Die Form für jedes Zeichen des gerade aktiven Zeichensatzes kann vom Benutzer pixelweise umdefiniert werden. Der folgende Befehl wählt das Zeichen an:

**ESC ESC 'J' a** Zeichen für Umdefinition auswählen.

Ein Zeichen wird durch 8 bzw. 16 Bytes definiert, die eine 8x8- oder 8x16-Matrix bilden (je nach Bildschirmauflösung). Die Bytes sind die Zeilen der Matrix, der oben links mit Bit 0 des ersten Bytes beginnt. Ein auf 1 gesetztes Bit erscheint am Bildschirm als heller Punkt.

Nach jeweils 8 (non-interlaced) bzw. 16 (interlaced) über Kanal 4 übertragenen Bytes ist das Zeichen fertig definiert, und es geht weiter mit dem darauffolgenden Zeichen. Das Zeichen 7Fh im aktuellen Zeichensatz ist zugleich das Füllmuster im Tektronix-Befehlssatz.

Beispiel: Zeichen '1' (8x8-Zeichenmatrix)

```
Byte 0: 00000000 (00h)
Byte 1: 00001000 (10h)
Byte 2: 00011000 (18h)
Byte 3: 00111000 (1Ch)
Byte 4: 00011000 (18h)
Byte 5: 00011000 (18h)
```

Byte 6: 00011000 (18h)  
 Byte 7: 00111100 (3Ch)

Beispiele: Für irgendeine exotische Anwendung soll anstelle des Zeichens 'A' (41h) ein Viereck, anstelle von 'B' (42h) ein Dreieck im Zeichensatz erscheinen.

Zunächst non-interlaced (8x8-Zeichenmatrix) im Direktmodus:

ESC ESC 'C4' (Zielkanal auf Zeichensatz)  
 ESC ESC 'JA' ('A' soll umdefiniert werden)  
 80h 80h FEh C2h C2h C2h FEh 80h 80h (Viereck; Direktmodus)  
 80h 80h 98h A4h C2h FEh 80h 80h 80h (Dreieck)  
 ESC ESC 'C0' (Zielkanal wieder Centronics)

Und jetzt das gleiche interlaced (8x16-Matrix) und mit String-Übertragung:

ESC ESC 'C4' (Zielkanal auf Zeichensatz)  
 ESC ESC 'JA' ('A' anwählen)  
 ESC ESC 'f' 20h 00h (String-Transfer, 32 Bytes)  
 00h 00h 00h 00h 7Eh 7Eh 42h 42h 42h (Viereck, obere Hälfte)  
 42h 42h 42h 7Eh 7Eh 00h 00h 00h 00h (Rest; danach ist 'B' dran)  
 00h 00h 00h 00h 18h 18h 24h 24h 42h (Dreieck, obere Hälfte)  
 42h 7Eh 7Eh 00h 00h 00h 00h 00h 00h (Rest des Dreiecks)  
 ESC ESC 'C0' (Zielkanal wieder Centronics)

Zu beachten ist, daß jetzt die Zeichen 'A' und 'B' in sämtlichen ASCII-Standardzeichensätzen '0'-'7' umdefiniert worden sind, jedoch nicht in den Index-Zeichensätzen. Dazu ist die Definition bei aktiver Indexschrift (-> ESC 'G') zu wiederholen.

### 5.7 Userprogramm und direkte Portzugriffe (Ziel/Quellenkanal 5)

Über Kanal 5 können eigene Befehle in GRIP's Befehlssatz implementiert werden. Dazu stehen insgesamt 256 Bytes ab Adresse 4300h für Routinen in Z80-Maschinencode zur Verfügung.

Bei der Wahl von Zielkanal 5 wird dieser Speicherbereich gelöscht, an die erste Adresse wird ein RET-Befehl (C9h) eingetragen. Jetzt können bis zu 256 Bytes Programmcode gesendet werden, die mit einem RET-Befehl enden müssen. Für den Stack stehen 40 Byte zur Verfügung.

Das Userprogramm wird durch einen der folgenden Befehle aufgerufen:

ESC ESC 'M' aa /bb      Userprogramm starten.

Die Hex/ASCII-Zahl aa wird beim Aufruf nach Z80-Register A übergeben, die Antwort bb wird nach Programmende aus Register A übernommen. Bei bb ist zur Unterscheidung von Tastaturdaten das 8. Bit der beiden Hex/ASCII-Zahlen gesetzt!

ESC ESC 'm' x            Userprogramm starten.

Das Byte x wird beim Aufruf nach Register A übergeben. Ein Antwort-Byte gibt es nicht.

Das Userprogramm kann direkt auf das RAM, den Bildschirmspeicher, auf Ports oder Peripheriebausteine zugreifen. Die Interrupts müssen mit DI abgeschaltet werden.

Unabhängig vom Userprogramm gibt es noch eine einfachere Möglichkeit, direkt auf I/O-Ports der GRIP zuzugreifen: die Portausgabebefehle.

ESC ESC 'O' aa bb      Direkte Portausgabe (ASCII/Hex). bb wird an den Port aa ausgegeben.

ESC ESC 'o' x y      Direkte Portausgabe. Das Byte y wird an den Port x ausgegeben.

*Userprogramm oder direkte Portausgabe sollten mit Sorgfalt verwendet werden. Durch fehlerhafte Programme kann GRIP zum 'Aussteigen' gebracht werden!*

### 5.8 Tasten-Umcodetabelle (Ziel/Quellenkanal 6)

Für jede gedrückte Taste kann statt des ASCII-Tastencodes ein beliebiger Zeichenstring zum Host gesendet werden. Auf diese Weise lassen sich Tasten umdefinieren oder Funktionstasten mit Befehlen belegen. Die Umcodierung erfolgt über eine Tabelle, die sich über Kanal 6 programmieren läßt und bis zu 1024 Bytes enthalten darf. Das Format ist:

```
Tastencode (1 Byte)
Stringlänge n (1 Byte)
String (n Bytes)
nächster Tastencode
usw. . . . . .
00h 00h (Ende der Tabelle)
```

Bei Stringlänge 0 wird die entsprechende Taste ausgeblendet, d.h. kein String gesendet. Zwei aufeinanderfolgende Nullbytes markieren das Ende der Tabelle.

Bei Anwahl von Kanal 6 wird die bisherige Umcodetabelle gelöscht, und es können bis zu 1024 Bytes gesendet werden. Die Zeichen, die im Tastaturpuffer in der Statuszeile erscheinen, sind die noch nicht umcodierten Originalcodes! Bei IBM-Tastaturen erfolgt die Umcodierung in ASCII vor der Umcodierung durch die Tabelle.

Beispiel: Aus einer amerikanischen QUERTY-Tastatur soll eine deutsche Tastatur gemacht werden. Dazu müssen die 'Y'- und 'Z'-Tasten vertauscht werden. Die Tastatur besitzt außerdem eine Funktionstaste, die bei Betätigung den Code 88h sendet. Beim Druck auf diese Taste soll das Directory auf den Bildschirm gebracht, also die Zeichen 'D', 'I', 'R', <CR> zum Host gesendet werden. Zur Verwirklichung des Ganzen sind folgende Befehle nötig:

ESC ESC 'C6'	(Zielkanal auf Umcodetabelle)
ESC ESC 'F'	(Hex-Übertragung einleiten)
'59015A5A0159'	(Y mit Z vertauschen)
'79017A7A0179'	(y mit z vertauschen)
'19011A1A0119'	(^Y mit ^Z vertauschen)
'88044449520D'	(88h sendet DIR <CR>)
'0000'	(Ende der Tabelle)
<CR>	(Ende der Hex-Übertragung)
ESC ESC 'C0'	(Zielkanal wieder auf Centronics)

### 5.9 Der Soundgenerator (Ziel/Quellenkanal 7)

Der Soundgenerator kann eine Folge von Tönen mit wählbarer Frequenz und Dauer abspielen. Die Tonfolge läßt sich über Kanal 7 programmieren und ertönt nach Übertragung des letzten Tons. Sie wird nach jedem BEL-Zeichen (07h) im TVI-Modus erneut abgespielt. Auf diese Weise läßt sich auch der BEL-Ton umprogrammieren.

Es können maximal 512 Töne im Soundpuffer gespeichert werden. Jeder Ton besteht aus 2 Bytes. Das erste Byte (b1) bestimmt die Tonhöhe F nach folgender Formel:

$$F = (117.2 \text{ kHz}) / (b1) \text{ für gerade } b1$$

$$F = (9.375 \text{ kHz}) / (b1) \text{ für ungerade } b1$$

Es lassen sich also Frequenzen zwischen 58.5 kHz ( $b1 = 2$ ) und 38 Hz ( $b1 = 255$ ) erzeugen.

Das zweite Byte b2 bestimmt die Dauer des Tons. Es ist folgendermaßen aufgebaut:

$$b2 = 001d4-d0$$

l: 0 = Pause

1 = Ton

d4-0 = Dauer in Einheiten von 20 Millisekunden.

Das Abspielen beginnt mit dem Empfang des Melodie-Endes, das wie bei der Umcode-Tabelle aus zwei aufeinanderfolgenden Nullen besteht. Nach Anwahl von Zielkanal 7 wird der Soundpuffer gelöscht (BEL sendet dann nur einen kurzen Glockenton) und ist für eine neue Melodie bereit.

Beispiel: Folgende Kommandos lassen eine kleine Melodie ablaufen:

ESC ESC 'C7'	(Zielkanal = Soundgenerator)
ESC ESC 'F'	(Hex-Übertragung einleiten)
'4230582458045824503058'	(Jetzt kommen die Töne)
'22681D4622460D4222420D'	
'0000'	(Ende der Melodie)
<CR>	(Ende der Übertragung)
ESC ESC 'C0'	(Zielkanal wieder Centronics)

### 5.10 Statuszeilen und Uhr (Zielkanal 9)

GRIP-6 kann zwei Statuszeilen auf dem Bildschirm darstellen: die *System-Statuszeile* am oberen und die *User-Statuszeile* am unteren Bildrand. Während in der System-Statuszeile neben der Uhr und dem Tastaturpuffer noch einige Symbole und Kennzahlen für den Betriebszustand angezeigt werden, läßt sich die User-Statuszeile vom Host aus mit einem beliebigen Text programmieren.

Anwählen von Zielkanal 9 löscht die User-Statuszeile. Es können dann bis zu 96 Zeichen in die Zeile übertragen werden. Sie bleiben im akkugepufferten RAM auch nach Abschalten der Spannung erhalten.

Die Statuszeilen lassen sich mit folgendem Befehl ab- oder umschalten:

ESC ESC 'S' a    Statuszeilen umschalten

- a = '0': Beide Statuszeilen aus.  
 '1': Nur System-Statuszeile anzeigen (default).  
 '2': Nur User-Statuszeile anzeigen.  
 '3': Beide Statuszeilen anzeigen.





### 5.11 Der Systempatch (Ziel/Quellenkanal 10)

Mit Hilfe des 'Systempatch'-Kanals kann der Host auf den Inhalt des statischen RAM direkt zugreifen. Dadurch lassen sich einzelne Betriebsparameter ändern.

Der Systempatch-Kanal ist ähnlich, aber 'allgemeingültiger' als die auf feste Adressen eingestellten Userprogramm- und Druckerpatch-Kanäle. Die Anfangsadresse wird mit folgendem Befehl eingestellt:

**ESC ESC 'p' aaaa** Setze RAM-Adresse für Systempatch.

Erlaubt sind Adressen von 4000h-5FFFh. Nach Anwahl der Adresse können über Kanal 10 ein oder mehrere Bytes übertragen werden. ACHTUNG - diese Funktion sollte mit Vorsicht benutzt werden!

### 5.12 Hardcopy und Druckerpatch (Ziel/Quellenkanal 11)

Mit dem Hardcopy-Befehl kann die am Bildschirm erstellte Grafik zu Papier gebracht oder zu Testzwecken ein Bildschirmausdruck erstellt werden. Dabei wird der sichtbare Inhalt des Video-RAM's Pixel für Pixel zum Drucker-Zielkanal - Centronics oder RS232 - übertragen. Die Übertragung wird durch folgenden Befehl ausgelöst:

**ESC ESC 'H'** Hardcopy starten: Bildschirminhalt komplett oder zeilenweise auf Grafik-Drucker ausgeben.

Dieser Befehl wird ignoriert, wenn ein anderer Zielkanal als Centronics, RS232 oder ECB-Bus eingestellt ist. Bei dem Drucker muß es sich um einen grafikfähigen Matrixdrucker (ein- oder mehrfarbig) handeln. Für Farbdruker ist die zeilenweise Hardcopy vorgesehen. Der Hardcopy-Modus wird durch folgenden Befehl eingestellt:

ESC ESC 'h' a Druckertyp für Hardcopy vorwählen, Zeilen-Hardcopy steuern.

- a = ' ': Zeilen-Hardcopy: Beginn mit erster Bildschirmspalte.  
'!': Zeilen-Hardcopy: Weiter mit nächster Spalte.  
'0': Gesamt-Hardcopy auf IBM/NL10, EPSON FX-80 o.ä.  
'1': Gesamt-Hardcopy auf STAR 10X oder kompatiblen.  
'2': Gesamt-Hardcopy auf Siemens PT-88.  
'3': Gesamt-Hardcopy auf NEC P7.  
'4': Gesamt-Hardcopy (Graustufen) auf NEC P6/P7.  
'5': Gesamt-Hardcopy (Hochauflösendes Querformat) auf NEC P6/P7.

Es lassen sich beliebige Matrix-Drucker mit 8 oder mehr Nadeln für eine Hardcopy anpassen. Dazu muß mit Hilfe des 'Druckerpatch'-Kanals ein Initialisierungsstring (IS) programmiert werden. Durch den <ESC> <ESC> 'h'-Befehl läßt sich dieser String für häufig verwendete Druckertypen vordefinieren. Bei der Gesamt-Hardcopy sendet GRIP den IS zum Zielkanal, danach folgt eine Anzahl von Bytes, die die Pixelinformation (Text und Grafik) der linken vertikalen Randspalte enthalten. Die Byte-Anzahl richtet sich nach dem im Tektronix-Modus eingestellten Y-Koordinatenbereich (-> <ESC> 'A'); defaultmäßig werden 560 Bytes (bzw. 6x560 Bytes beim Graustufenausdruck) übertragen. Dieser Ausdruck wird Spalte für Spalte wiederholt, bis der gesamte Bildschirminhalt (um 90 Grad gedreht) ausgedruckt ist. Bei Verwendung der XR-Karte kann die auszudruckende Bit-Ebene des Bildschirmspeichers mit <ESC> <ESC> 'v' vorgewählt werden. Ein helles Bildschirmpixel erscheint normalerweise schwarz auf dem Ausdruck. Graustufen-Hardcopy (a = '4') ist nur von der XR-Karte und mit einem 24-Nadel-Drucker möglich. Dabei ist jedem Bildschirmpixel eine 2x2-Pixelmatrix auf dem Drucker zugeordnet, die nach einem Zufallsalgorithmus gemäß dem Grauwert (Farbindex) des Pixels schattiert wird.

**ACHTUNG:** Da die Hardcopy aus Kompatibilitätsgründen zur Vorgänger-Karte GRIP-4 auf 560 Zeilen vertikal voreingestellt ist, muß bei anderen Werten - wenn mit ESC 'A' (s.o.) im Tektronix-Modus die Bildhöhe verändert wurde - der Initialisierungsstring auf die neue Zeilenzahl eingestellt werden! Genaueres entnehmen Sie bitte Ihrem Druckerhandbuch. Die zeilenweise Hardcopy ist zur Grafikausgabe auf einen Farbdrucker vorgesehen. Hierbei kann jede Bildschirm-Spalte (= Druckerzeile, 8 Pixel breit) mehrmals übereinander gedruckt werden, wobei nach jedem Druckvorgang die Farbe und zugleich die Bit-Ebene des Bildschirmspeichers gewechselt wird.

<ESC><ESC>'h' initialisiert die Ausgabe auf die Spalte am linken Bildschirmrand, <ESC><ESC>'h!' wählt die jeweils nächste Spalte zum Ausdrucken vor. Der IS muß einen Zeilenvorschub um 8 bzw. 24 Nadelbreiten und einen Wagenrücklauf auslösen und anschließend den Drucker so auf Grafikausgabe initialisieren, daß die folgenden Bytes in korrekter Anzahl als Bitmuster in einer Zeile abgedruckt werden. Bei spaltenweiser Hardcopy entfällt der Zeilenvorschub.

Der IS beginnt mit einem Steuerbyte, danach folgt die Druckerinitialisierung (14 Bytes); den Schluß bildet ein weiteres Steuerbyte für 24-Nadel-Drucker. Die Steuerbytes haben folgenden Aufbau:

S D Z T L3 L2 L1 L0 (1. Byte)

0 0 0 0 0 IV DD GR (16. Byte)

S = 0: Grafik-Bytes werden gespiegelt gesendet.  
1: Grafik-Bytes werden direkt gesendet.

D = 0: Nach jeder Bildschirmzeile folgt eine Leerzeile.  
1: Jede Bildschirmzeile wird zweimal gedruckt.  
Im Interlaced-Modus hat das d-Bit keine Bedeutung.

T = 0: Normales Bildformat 768x576 Pixel.  
1: Gedehtes Bildformat 768x840 Pixel: Jede zweite Bildschirmzeile wird zweimal gedruckt.

Z = 0: Gesamt-Hardcopy.  
1: Zeilenweise Hardcopy.

L3-L0: Länge des folgenden Druckerstrings (1-14 Bytes).

DD = 0: Doppelte Dichte aus (8-Nadel-Betrieb).  
1: Doppelte Dichte ein (24-Nadel-Betrieb).

GR = 0: Graustufen aus.  
1: Graustufen ein (XR-Ausdruck, 24 Nadeln).

IV = 0: Ausdruck invertiert.  
1: Ausdruck nichtinvertiert.

Einige der vordefinierten IS mit Steuerbyte:

EPSON FX-80: 09 1B 4A 18 0D 1B 2A 05 30 02 00 00 00 00 00  
STAR: 18 1B 4A 10 0D 1B 4C 30 02 00 00 00 00 00 00  
SIEMENS PT-88: 89 0D 0A 1B 5B 30 35 36 30 79 00 00 00 00 00

**Achtung:** Vor Ausführung der Hardcopy mit Siemens PT-88 muß der Drucker mit <ESC>'[2w' auf 12 CPI und mit <ESC>'[08x' auf 8/72 Zoll Zeilenvorschub geschaltet werden.

## 6 Spooler und zweite Bildschirmseite

Die insgesamt 196 KByte Video-Speicher der GRIP-6B sind in drei Ebenen aufgeteilt, von denen sich jede - je nach Bildschirmauflösung - ggfs. noch einmal in zwei Seiten (Vorder- und Hintergrundseite) teilen läßt. Auch die drei Ebenen können wahlweise parallel (8 Farben) oder als einzelne Seiten (monochrom) am Bildschirm dargestellt werden. Im Non-Interlaced-Modus wird nur ein 32-KByte-Bereich des Bildschirmspeichers zur Bilddarstellung benötigt. Die zweite Hälfte der 64-KB-Video-RAM-Ebene ist frei. In der Ebene 0 kann sie zusätzlich wahlweise als Druckerspooler benutzt werden. Die Ebenen und Seiten werden durch den folgenden Befehl umgeschaltet:

ESC ESC '5' a    Bildschirmseiten zuordnen.

- a = '0': Schreiben in Vordergrundseite, Spooler ein.  
'1': Schreiben in Vordergrundseite, Bild dunkel, Spooler ein.  
'2': Schreiben in Vordergrundseite, Bild dunkel, Spooler aus.  
'3': Schreiben in Vordergrundseite, Spooler aus (default).  
'4': Schreiben in Hintergrundseite, Spooler aus.  
'5': Darstellung Vorder- und Hintergrundseite vertauschen.

*Die Parameter '0'-'5' sind nur non-interlaced wirksam.*

- 'a': Darstellung nur Ebene 0 monochrom (nur GRIP-6B)  
'b': Darstellung nur Ebene 1 monochrom (nur GRIP-6B)  
'c': Darstellung nur Ebene 2 monochrom (nur GRIP-6B)  
'd': Darstellung Ebenen 0-2 achtfarbig (default).  
'A': Schreiben nur in Ebene 0 (nur GRIP-6B)  
'B': Schreiben nur in Ebene 1 (nur GRIP-6B)  
'C': Schreiben nur in Ebene 2 (nur GRIP-6B)  
'D': Schreiben in Ebenen 0-2 (Farbmodus, default).

Der Host kann die Hintergrundseite "unsichtbar" beschreiben, während auf dem Bildschirm die Vordergrundseite sichtbar ist. Diese

Betriebsart wird durch <ESC><ESC>'54' eingestellt. Nach <ESC>-<ESC>'55' werden die Seiten vertauscht, so daß die bisherige Hintergrundseite dargestellt und die bisherige Vordergrundseite zur Hintergrundseite wird. Vorder- und Hintergrundseite sind unabhängig; das Scrollen erfolgt getrennt, und die Cursorpositionen bleiben auch beim Umschalten erhalten.

Bei GRIP-6B lassen sich die drei 64KB-Ebenen im Monochrom-Modus umschalten, so daß maximal sechs Bildschirmseiten zur Verfügung stehen, die getrennt beschrieben und angezeigt werden können. Der 32-KByte-Spooler liegt in der Hintergrundpage der Ebene 0 und dient der Zwischenspeicherung der Daten vom Host zum Drucker. Er faßt etwa 10-15 Textseiten und erlaubt es, während des Ausdrucks gleichzeitig Daten an den Bildschirm auszugeben. Das Drucken kann durch folgenden Befehl vorzeitig abgebrochen werden:

ESC ESC 'P'      Spooler löschen.

## 7 Lichtgriffel und Statusabfrage

Der Lichtgriffel-Detektor reagiert auf eine HIGH-Flanke am LPEN-Eingang. Der Detektor wird über folgenden Befehl abgefragt:

**ESC ESC 'L' /x a b** Lichtgriffel-Position abfragen.

x = Rasterzeilennummer + A0h

a = Block-Nummer vertikal + A0h

b = Block-Nummer horizontal + A0h

Je nach Ansprechgeschwindigkeit des Lichtgriffels kann die Position in horizontaler Richtung etwas verschoben sein. Dies muß durch das Anwendungsprogramm korrigiert werden. Die Blocknummern entsprechen der Pixelposition, geteilt durch 8. Die Nummern (0,0) entsprechen der oberen linken Ecke auf dem Bildschirm; die Statuszeile ist dabei mit einbezogen. Bei den drei Antwort-Bytes ist das 8. Bit gesetzt!

Zum Abfragen des Video-Controllers gibt es noch den Befehl:

**ESC ESC '\$' /x** Controller-Status abfragen.

x = 1 0 RY 0 SY EX EF VB

VB = Vertikal Blanking (vertikale Austastlücke)

EF = Even Field (erstes Halbbild Interlaced)

EX = Externes BAS-Video-Signal angeschlossen.

SY = GENLOCK einsynchronisiert

RY = Bild fertig eingelesen

Die Flags EX,SY,RY sind nur in Verbindung mit einer XR-Genlock-Karte sinnvoll. Das RY-Flag wird nach der Statusabfrage automatisch zurückgesetzt.

## 9 Bildschirmformate

Die folgenden Befehle dienen der Wahl der Darstellungsart, des Textformats und der Monitoreinstellung.

ESC ESC 'r' a Betriebsart umschalten.

a = '0': Non-Interlaced-Modus (default).

a = '1': Interlaced-Modus.

Bei diesem Befehl wird gleichzeitig der Bildschirm gelöscht. Im Interlaced-Modus sind Spooler und zweite Bildschirmseite abgeschaltet, ein nachleuchtender Monitor sollte verwendet werden. Die vertikale Auflösung verdoppelt sich auf 576 Bildpunkte.

ESC ESC '6' a Zeilenabstand und Smooth Scroll einstellen.

a = 0 s1 s0 d z3 z2 z1 z0 (default 00111000)

s1-0 = 00 -> Hard Scroll

01 -> Auto Smooth Scroll (default)

11 -> Standard Smooth Scroll

d = 1 -> Normal Size, 0 -> Double Size (s.u.)

z0-3 = Zeichenhöhe, Wertebereich 7..11 Pixel/Zeichen

Dieser Befehl ist im Interlaced-Modus inaktiv. Durch Verändern der Zeichenhöhe (im Interlaced-Modus immer 16 Pixel) können die Textzeilen zur besseren Lesbarkeit auseinandergezogen werden; der Zwischenraum bleibt dunkel. Im Grafikbetrieb (Vektorzeichnen) muß der Zeilenabstand auf 8 Pixel eingestellt sein.

Mit Zeichenhöhe 7 lassen sich bis zu 41 Zeilen am Bildschirm darstellen. Dazu sollte sinnvollerweise die Indexschrift gewählt werden (<ESC>'G'^P). Der Befehl zum Erweitern des Textformats (<ESC><ESC>'8') ist für über 36 Zeilen nur gültig, wenn vorher 7 Pixel Zeichenhöhe eingestellt wurden. Danach sollte ein Bildschirmlöschen (^L) erfolgen. Die Zeichenhöhe 11 entspricht dem Zeilenabstand normaler Terminals mit 25 Zeilen/ Bildschirm.

Bei Smooth Scroll (pixelweise gleitendes Rollen) rollt entweder der ganze Bildschirm oder nur der untere Teil (beim Zeilenlöschen). Durch dynamische Veränderung der Rollgeschwindigkeit (Auto Smooth Scroll) wird, im Gegensatz zu vielen Terminals mit dieser Eigenschaft, das Scrollen nicht langsamer. Soll das Scrollen immer mit konstanter Geschwindigkeit erfolgen, so ist das Auto Scroll Feature mit dem 's1'-Bit abzuschalten. Der Smooth Scroll funktioniert auch unter WORDSTAR oder TURBO-PASCAL und läßt sich über das 's0'-Bit ganz abschalten. Er ist im Interlaced-Modus inaktiv.

Double Size (doppelte Zeichenhöhe für den gesamten Bildschirm) läßt sich über das d-Bit einschalten. Durch die doppelte Zeichenhöhe verringert sich die Anzahl der verfügbaren Zeilen auf die Hälfte.

Beispiel: <ESC><ESC>'69' (1Bh 1Bh 36h 39h) stellt Auto Smooth Scroll und 9 Pixel/Zeile ein.

**ESC ESC '8' a b** Textformat wählen.

a = (Zeilenzahl-1) + 20h (default: 37h = 24 Zeilen)

b = (Spaltenzahl-1) + 20h (default: 6Fh = 80 Spalten)

Hiermit wird die Zahl der Zeilen und Spalten auf dem Bildschirm eingestellt. Bei Breitschrift (s.u.) ist die Spaltenzahl halbiert. Wenn mit <ESC><ESC>'6' ein anderer Zeilenabstand eingestellt wurde, ändern sich die Maximalwerte für Zeilen- und Spaltenzahl dementsprechend.

Beispiel: <ESC><ESC>'8C' (1Bh 1Bh 38h 43h 7Fh) wählt das Format von 96x36 Zeichen (dazu müssen beide Statuszeilen abgeschaltet sein).

ESC ESC '9' x Video-Signal an Monitor anpassen.

x = 0 i w1 w0 h1 h0 v1 v0

i = 0: Non- oder Normal-Interlaced (default)  
1: Pseudo-Interlaced

w1-0 = 00: Kein Horizontal-Syncimpuls  
01: Horizontal-Sync kurz  
10: Horizontal-Sync mittel (default)  
11: Horizontal-Sync lang

h1-0 = 00: Bild nach links  
01: Bildlage normal (default)  
10: Bild nach rechts  
11: Bild stark nach rechts

v1-0 = 00: Bild nach oben  
01: Bildlage normal (default)  
10: Bild nach unten  
11: Bild stark nach unten

Der Pseudo-Interlaced-Modus erhöht scheinbar die vertikale Auflösung durch Verdoppeln jeder Pixelzeile, ist aber nur für nachleuchtende Monitore (P39-Phosphor) zu empfehlen. Beim Anschluß einer Genlock-Erweiterung muß dieser Modus allerdings in der Regel eingestellt sein, da die meisten Kameras interlaced arbeiten. Wenn die Zeichen in der obersten Zeile

verzerrt werden, ist das Bild nach unten zu verschieben ( $v1-0 = 10$  oder  $11$ ).

Beispiel: `<ESC><ESC>'9e'` (1Bh 1Bh 39h 65h) schaltet bei normaler Bildlage den Pseudo-Interlaced-Modus ein.

**ESC ESC 'Q'**      Monitor-Testbild einschalten.

Das Testbild füllt den Bildschirm mit Einstellmarken und die aktuellen Zeichensätze. Es erleichtert die optimale Anpassung des Monitors mit dem Befehl `<ESC><ESC>'9'`.

**ESC ESC 'R'**      Reset-Befehl: Bildschirm und alle Puffer löschen, alle Funktionen auf die Default-Werte zurücksetzen.

Achtung - der Reset-Befehl löscht als einziger Befehl auch den Hostpuffer! Deswegen werden die unmittelbar folgenden Befehle, wenn sie schon im Puffer stehen, nicht mehr ausgeführt. Es empfiehlt sich, nach Senden des Reset-Befehls eine Verzögerung (ca. 500 ms) einzubauen.

## 10. Das Set-Up-Menü

In diesem Menü lassen sich bestimmte Parameter permanent einstellen; sie werden nichtflüchtig im EEPROM abgespeichert. Das Menü wird über die Tastatur aufgerufen: Nach Betätigen der Set-Up-Taste (00h bzw. ^@ bzw. <End> bei PC-Tastaturen oder <Pause> bei Siemens AT Tastatur) erscheint es auf dem Bildschirm.

Alternativ kann der Aufruf vom Hostrechner aus erfolgen, und zwar über den Befehl

ESC ESC '+'      Set-Up Modus aktivieren.

Durch <SPACE> und <BACKSPACE> wird das jeweilige Untermenü gewählt. Die Menüpunkte werden dann durch <RETURN> aufgerufen. Das Verändern der Parameter erfolgt zyklisch durch <SPACE> (vorwärts) und <BACKSPACE> (rückwärts), die Auswahl wieder durch <RETURN>. Mit der Set-Up-Taste oder mit <ESC> wird das jeweilige Untermenü verlassen, wobei die eingestellten Parameter übernommen werden. Die Parameter werden durch Wahl von <Install> resident im EEPROM abgespeichert. Mit <Initial> kann der Ursprungszustand wiederhergestellt werden. Mit <Exit> wird der Set-Up Modus verlassen.

Zu beachten ist, daß <Install> sämtliche Parameterwerte der Karte als Default-Parameter abspeichert, also auch den gewählten Zeichensatz, die Textfarbe etc.

### **Untermenü <Host interface>:**

Hier wird der Anschluß des Hostrechners (seriell, ECB-Bus oder Local, d.h. Tastatur) eingestellt, ebenso die Baudrate und das RS232-Protokoll. Im Local Mode erfolgt keine Stringkonvertierung.

Das Sonderfeature "Input Code Conversion" (nur GRIP-7) bewirkt, daß Zeichen mit gesetztem 7. Bit als Sonderzeichen auf dem Display angezeigt werden. Ist die Input Code Conversion

ausgeschaltet, so werden Zeichen mit gesetztem 7. Bit zum Zielkanal gesendet (s. -> Kanäle) und erscheinen nicht auf dem Display.

#### **Untermenu <Keyboard interface>:**

Einstellen von Tastatur-Art, -Baudrate und -Parametern. Es kann zwischen PC/AT- und serieller Tastatur gewählt werden.

#### **Untermenu <Screen>:**

Einstellen von Bildschirmformat, Scroll-Modus und Statuszeilen.

#### **Untermenu <Colours>:**

Einstellen der Look-Up-Tafel, Text-, Hintergrund- und Blinkfarben. Die Red/Green/Blue-Menupunkte verändern die mit "Colour" gewählte Farbe. Die zuletzt eingestellte Farbnummer wird beim Abspeichern der Parameter als Default-Textfarbe übernommen.

#### **Untermenu <Time intervals>:**

Einstellen der Blinkrate von Farben und Cursor, der Tondauer für das <BEL>-Signal sowie der Screen-Off-Automatik. Die Screen-Off-Zeit ist auf die Zeit (in Minuten) nach dem letzten übertragenen Zeichen bis zum automatischen Dunkelschalten des Bildschirms, um die Bildröhre zu schonen. Bei dem eingestellten Wert "0" ist die Automatik inaktiv.

**Untermenu <Pattern>:**

Das Testbild wird eingeschaltet.

**Untermenu <Monitor Adjust>:**

Hiermit läßt sich GRIP an das Display anpassen. Bei Anwahl des Menüpunktes erscheint das Testbild. In dem Parameterfeld in der oberen linken Ecke lassen sich 'on-line' die folgenden Werte verstellen:

VERT FREQ - Bildfrequenz (Bildwiederholrate) in Hz. Der hier einstellbare Wert ist abhängig von der Zeilenfrequenz (HOR FREQ) und von der Anzahl der Textzeilen (LINES). Je geringer die Zeilenzahl, desto höher kann die Bildfrequenz gewählt werden. Umgekehrt lassen sich bei entsprechend niedriger Bildfrequenz bis zu 80 Textzeilen am Bildschirm darstellen.

HOR FREQ - Zeilenfrequenz in kHz. Der hier einstellbare Wert ist abhängig von der Anzahl der Zeichen pro Zeile (COLUMNS). Mit Verändern dieses Wertes ändert sich automatisch auch die Bildfrequenz - es ist daher sinnvoll, zuerst die Anzahl der Zeilen und Spalten, dann die Zeilen- und dann die Bildfrequenz einzustellen, wenn bestimmte Frequenzen gefordert werden.

VERT POS, HOR POS - Lage des horizontalen und vertikalen Sync-Impulses (in Einheiten von Spalten bzw. Zeilen ab Bildschirmbeginn). Hiermit läßt sich das Bild auf dem Monitor verschieben.

LINES - Anzahl der Textzeilen. Bei Verändern dieses Wertes ändert sich automatisch auch die Bildfrequenz (VERT FREQ).

**COLUMNS** - Anzahl der Zeichen pro Zeile. Bei Verändern dieses Wertes ändert sich ggfs. automatisch die Bild- und Zeilenfrequenz. Es lassen sich 40..96 Zeichen pro Zeilen einstellen.

**LINE SPACE** - Zeilenabstand in Pixeln (7..11). Zwischen den Textzeilen werden Leerabstände eingefügt, und die Bildfrequenz verändert sich. Im Grafikmodus und bei 16-Pixel-Zeichensätze muß hier der Wert 8 stehen, da sonst Linien und Zeichen am Bildschirm unterbrochen erscheinen.

**LINE HEIGHT** - Zeichenhöhe in Pixeln (8 oder 16). Bei einer Veränderung dieses Wertes wird die Zeilenzahl (LINES) automatisch halbiert bzw. verdoppelt.

**INTERLACE** - einstellbar zwischen on/off/pseudo. Im Interlaced-Modus (Zeilensprungverfahren) läßt sich bei gleicher Bildfrequenz die vertikale Zeilenzahl verdoppeln, dafür flimmert das Bild stärker. Ab einer bestimmten Zeilen- und Spaltenzahl läßt sich Interlace nicht mehr auf 'on' stellen, da der Bildschirmspeicher nicht mehr ausreicht. Im Gegensatz zu den anderen Parametern wird der neue Modus erst wirksam, wenn das SETUP verlassen wird.

**2ND PAGE** - einstellbar zwischen on/off. Wenn mehr als 36 Charakterzeilen (= 288 Pixelzeilen) erforderlich sind, muß die Hintergrundpage abgeschaltet und dem Bildschirmspeicher zugeordnet werden.

### Untermenu <System Flags>

Im Menüpunkt <Setup Code> läßt sich der Tastaturcode (1 Byte) einstellen, mit dem von der Tastatur das Setup-Menü aufgerufen wird. Normalerweise ist dies der Code 00h.

-----ooOoo-----

# GRIP-6

## Board-Grafikterminal

### Hardware Installation

Copyright © CONITEC 1989, 1990, 7/1991

Hardware: Johannes C. Lotter  
Layout: Gesa Emde  
Software: Bernhard Emese /Thorsten Siebeking  
Handbuch: Johannes C. Lotter

Kein Teil dieser Dokumentation darf ohne ausdrückliche schriftliche Genehmigung von CONITEC reproduziert, vervielfältigt, gespeichert oder übersetzt werden. Für die Richtigkeit der hier angegebenen Daten übernehmen wir keine Haftung. Änderungen, die dem Fortschritt dienen, behalten wir uns vor, auch ohne dies besonders anzukündigen.

 **CONITEC DATENSYSTEME**

6100 Darmstadt • Grafenstr. 31 • ☎ (06151) 26013 • Fax (06151) 26015

## 1

**Einführung**

Die Board-Terminals der GRIP-Serie sind für Industrieanwendungen – Maschinensteuerung, Bedienerführung oder Prozeßvisualisierung – ausgelegt. Sie werden mit Standard-Befehlscodes über eine serielle Multinorm-Schnittstelle oder parallel (als I/O-Port am ECB-Bus) angesteuert. Jedes Terminalboard enthält ein Bildschirminterface für Text und Grafik sowie verschiedene Schnittstellen für Hostrechner, Tastatur und Drucker.

GRIP-6A stellt Text und Grafik monochrom dar, GRIP-6B in acht Farben oder Graustufen, GRIP-6C kann Text über eine Look-Up-Tafel in 128 Farben aus 262.144 Farbtönen darstellen. Die A- und B-Version läßt sich durch weitgehenden CMOS-Aufbau, erweiterten Temperaturbereich und den Watchdog-Timer auch in rauher, störverseuchter Umgebung einsetzen.

## 1.1

**Technische Daten**

- Emuliert TVI950, VT100 und Tx4010-Terminals
- Textdarstellung bis 96 Zeichen in 36 Zeilen
- Textaufbau mit 5000 Zeichen/sec.
- Hard- und Softscroll, umschaltbar
- 9 interne und 2 ladbare Zeichensätze
- Grafik bis 768 x 576 Bildpunkte
- Zeichnet Linien, Vektoren und Flächen
- Serielle oder AT-Tastatur anschließbar
- Parallel-Port (8 Bit) für LEDs oder Drucker
- Hostrechner-Anschluß über ECB-Bus, V24 oder RS422
- Watchdog-Timer für Betriebssicherheit
- Set-Up-Menue, Parameter werden im EEPROM abgespeichert
- Bild- und Zeilenfrequenz per Set-Up-Menue einstellbar
- Platine im Einfach-Europaformat (100x160 mm)

## 1.2 Vergleichstabelle

	GRIP-6A	GRIP-6B	GRIP-6C
Bildschirmseiten	2	2 bzw. 6	2
Bildpunkttakt	15 MHz	15 MHz	15 MHz
Bildfrequenz	50..70 Hz	50..70 Hz	50..70 Hz
Zeilenfrequenz	15..24 kHz	15..24 kHz	15..24 kHz
Text-Emulation	VT100, TVI950	VT100, TVI950	VT100, TVI950
Text-Farben	nein	8 von 8	128v.262.144
Text-Blinken	nein	ja	ja
Breitschrift	ja	ja	ja
Hochschrift	ja	ja	ja
Zeichensätze	9+2	9+2	9+2
Grafik-Emulation	TX 4010	TX 4010	TX 4010
Auflösung (max)	768x576	768x576	768x576
Grafik-Farben	nein	8 von 8	2 v. 262.144
Vektoren	ja	ja	ja
Flächenzeichnen	ja	ja	ja
Zeichensatz-RAM	4 kByte	4 kByte	4 kByte
Video-RAM	64 kByte	192 kByte	64 kByte
Attribut-RAM	--	--	64 kByte
Befehlspeicher	512 Byte	512 Byte	512 Byte
Umcode-Tabelle	1 KByte	1 KByte	1 KByte
Setup-EEPROM	ja	ja	ja
Watchdog-Timer	ja	ja	ja
Look-Up-Tafel	--	74S189	G176
Bildschirm-Anschl.	BAS/TTL, s/w	Farbe (TTL)	Farbe
Host-Anschluß	ECB/V24/422	ECB/V24/422	ECB/V24/422
Tastatur-Anschluß	TTL/V24/AT	TTL/V24/AT	TTL/V24/AT
Drucker-Anschluß	V24/Centron.	V24/Centron.	V24/Centron.
Maße (o.Stecker)	100x160 mm	100x160 mm	100x160 mm
Stromversorgung	5V/450mA	5V/600mA	5V/700mA
Temperaturbereich	-20..+80°C	-20..+80°C	0..+70°C

### 1.3

### Funktionen

GRIP emuliert im **Textmodus** die Befehlssätze der Terminals VT100 und TVI950. Durch das transparente Video-RAM ist das Bild auch bei Scrollen, Zeileneinfügen usw. stabil. Der Bildaufbau geht sehr schnell vor sich (Grenze ca. 50.000 Baud). Die Zeichen lassen sich in verschiedenen Größen und Farben und mit einstellbarem Textformat (40x24 bis 96x36) darstellen.

Der Bildspeicher stellt zwei Bildschirmseiten in der normalen Auflösung (768x288 Bildpunkte) zur Verfügung, in doppelter Auflösung nur noch eine Seite. Bei Verzicht auf Farbe lassen sich auf GRIP-6B bis zu 6 Bildschirm-Seiten definieren. In den nicht sichtbaren Bildschirmseiten kann Text 'im Hintergrund' aufgebaut werden.

GRIP enthält 9 Standard-Zeichensätze, die einfach, invertiert, mit doppelter Breite oder doppelter Höhe dargestellt werden können. Zwei weitere Zeichensätze können vom Benutzer definiert werden und bleiben bei Anschluß einer Pufferbatterie auch nach dem Abschalten erhalten.

Der Zeilenvorschub erfolgt wahlweise hart oder weich (Soft Scroll). In der (abschaltbaren) System-Statuszeile am oberen Bildrand wird der Betriebszustand der Karte angezeigt. Außerdem enthält die Statuszeile noch eine Uhr und ein Tastaturfenster, das einen Blick auf die letzten 15 Bytes im Tastaturpuffer erlaubt. Für Menüs gibt es eine zweite User-Statuszeile am unteren Bildrand, die vom Benutzer beschrieben werden kann.

Die **Grafik** (Auflösung 768x576 Bildpunkte) wird durch leistungsfähige Kommandos unterstützt (Vektorzeichnen, Spray-Effekte, Block-, Muster- und Flächenfüllen). Die Standard-Software emuliert ein Tektronix-Grafikterminal; neben dem Vektormodus sind noch Alpha-Textmodus, Punktmodus und ein plotterähnlicher Inkrementalmodus implementiert.

GRIP-6B unterstützt Farbgrafik in acht Farben. Für spezielle Zwecke kann der Hostrechner auch direkt (bytwweise) auf den Bildspeicher zugreifen.

Zur **Dateneingabe** können serielle (V24 oder TTL) oder PC/AT-Tastaturen angeschlossen werden. Die letzten 64 Eingaben werden im Tastaturpuffer zwischengespeichert, die letzten 15 in der Statuszeile angezeigt. Über eine 1000 Zeichen große Umcode-Tabelle lassen sich Funktionstasten mit beliebigen Zeichenfolgen belegen.

Eine zweite **V24-Schnittstelle** ist zum Anschluß z.B. eines Druckers oder Modems an den Hostrechner vorgesehen. Baudraten, Datenformat und Protokolle (RTS/CTS oder XON/XOFF) sind vom Host oder per Set-Up-Menü programmierbar.

Für das Druckerinterface steht zusätzlich noch die **Centronics-Schnittstelle** zur Verfügung. Alternativ kann diese Schnittstelle als 8-Bit-Universalport verwendet werden, z.B. für LEDs, die vom Host aus geschaltet werden.

Für die Datenübertragung zum Drucker läßt sich eine Bildschirmseite als Zwischenspeicher (Spooler) umdefinieren, der ca. 10 bis 15 DIN-A4-Seiten Text faßt. Der Inhalt des Bildspeichers kann mit dem Hardcopy-Befehl in voller Auflösung zu Papier gebracht werden.

Der **Tongenerator** erzeugt Signale mit einer programmierbaren Melodiefolge. Sein Speicher faßt 512 Töne. Der Lautsprecherausgang wird auch für das ASCII-Glockensignal (BELL) und für den Tastenklick benutzt.

## 2

## Inbetriebnahme

An GRIP lassen sich unterschiedliche Displays, Tastaturen und Host-Schnittstellen anschließen. Die Konfiguration erfolgt über das **Set-Up-Menu** (s. Befehlssatz). Die Set-Up-Konfiguration wird im EEPROM resident installiert und steht danach immer zur Verfügung.

Da aber zum Bedienen des Set-Up-Menus bereits eine Tastatur konfiguriert sein muß, ist ein Jumperfeld (J7) zum Voreinstellen von Tastatur und Hostrechner vorgesehen. *Steckt auf diesem Feld ein Jumper, so wird die im EEPROM eingestellte Konfiguration ignoriert, und die Bild- und Zeilenfrequenz des Monitors ist fest auf 50 Hz / 15.625 kHz eingestellt.*

In das Set-Up-Menu gelangen Sie am einfachsten, indem Sie per Jumper J7 die angeschlossene Tastatur wählen und den **Local-Modus** einstellen (Jumper-Pos. J7 3-5, s.u.). Dann läßt sich mit der Tastenkombination <ESC><ESC>'+' das Set-Up-Menu aufrufen. Nach Installieren der Konfiguration können die J7-Jumper entfernt werden, so daß von da an die Karte beim Einschalten ihre Konfigurationsdaten aus dem EEPROM übernimmt.

Vor Inbetriebnahme sind die Default-Positionen der Jumper zu kontrollieren (s. Anhang). Beim Einschalten der Betriebsspannung erfolgt ein Selbsttest der Karte; dabei muß die rote LED kurz aufleuchten (1/2 sec.) und dann erlöschen. *Leuchtet die LED stetig oder blinkt sie, so liegt ein Hardware-Fehler vor: Zu geringe Betriebsspannung (unter 4.75 Volt) oder Kurzschluß eines Signals an einem Stecker!*

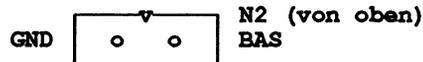
Einige Konventionen bei den folgenden Beschreibungen: Hex-Werte sind mit einem nachgestellten 'h' gekennzeichnet (z.b. C0h). Die Stifte der Jumper und Pfostenstecker sind gleichartig numeriert; Pin 1 ist im Bestückungsplan (Anhang) mit einem Dreieck gekennzeichnet.

Im Anhang finden Sie die Schaltpläne und die Steckerbelegungen. Die GRIP-6B-Version ist dort als 'GRIP-6', die Varianten GRIP-6A und GRIP-6C als 'GRIP-7' bezeichnet.

## 2.1 Anschluß des Monitors

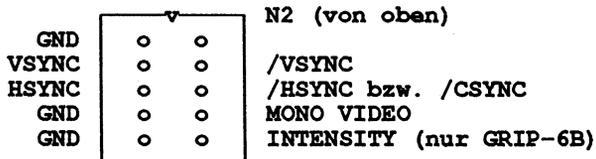
### 2.1.1 Monochromer BAS-Monitor an GRIP-6A/6B/6C

Der BAS-Monitor wird an N2, Pin 2 angeschlossen. Bei GRIP-6B kann der Video-Pegel (an 75 Ohm Last) mit Poti P1 zwischen 0.5 und 3.5 Volt eingestellt werden; bei GRIP-6A/6C ist der Pegel auf 1 Volt fest eingestellt.



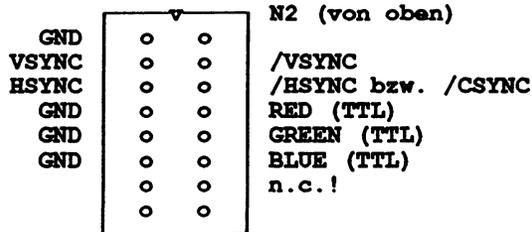
### 2.1.2 Monochromer TTL-Monitor (sep. Sync) an GRIP-6A/6B

Anschluß bei negativen Sync-Signalen an N2, Pins 4 (/VSYNC), 6 (/HSYNC bzw. /CSYNC) und 8 (VIDEO). Positive Sync-Signale können an den Pins 1 (VSYNC) und 3 (HSYNC) abgenommen werden. Bitte beachten Sie, daß bei GRIP-6A die Verbindungsbrücke zwischen Pins 1 und 12 bei Z39 (28pol. Sockel) gesteckt sein muß!



### 2.1.3 RGB-Farbmonitor an GRIP-6B

Der Monitor wird an Stecker N2 angeschlossen. Die Sync-Signale stehen in jeder gewünschten Polarität zur Verfügung. Der /CSYN-Ausgang kann mit dem /CSYNC- oder /HSYNC-Eingang des Monitors verbunden werden; im letzten Fall muß auch /VSYN mit dem /VSYNC-Eingang des Monitors verbunden werden.



Es lassen sich auch RGB-Analogmonitore (z.b. Multisync) anschließen. In diesem Falle müssen in die RED-, GREEN- und BLUE-Zuleitungen 270-Ohm-Widerstände in Serie eingefügt werden, um das TTL-Signal auf Analogpegel (1V) umzusetzen. Die Widerstände bilden Spannungsteiler mit den 64-Ohm-Eingangswiderständen des Monitors.

Zum Anschluß eines 50-Hz-Studiomonitors mit R/G/B/CSYNC-Eingang an GRIP-6B ist auch in die /CSYNC-Zuleitung ein 270-Ohm-Widerstand einzufügen.

*ACHTUNG* - Bei Anschluß eines Flachbandkabels an N2 darf das CKPIX-Signal (Pin 14) nicht angeschlossen werden! Dieses Signal ist nur für Aufsteck-Platinen und EL-Displays gedacht und wird von Leitungslängen über 15 cm kapazitiv überlastet.

### 2.1.4 RGB-Analog-Farbmonitor an GRIP-6C

Anschluß an N2, wie im vorigen Abschnitt beschrieben. Da die R/G/B-Signale bei GRIP-6C bereits mit 1-V-Pegel zur Verfügung stehen, entfällt eine Umsetzung mit Widerständen. Die Farbpegel lassen sich mit Poti P1 einstellen.

Ein Beispiel für den Anschluß an einen Multisync-Monitor mit 9- oder 15-poliger Buchse:

GRIP-6C, N2	9-polig (NEC)	15-polig (VGA)
8 (RED) ----->	1 (RED)	1 (RED)
10 (GREEN) ----->	2 (GREEN)	2 (GREEN)
12 (BLUE) ----->	3 (BLUE)	3 (BLUE)
6 (/CSYN) ----->	4 (Sync.)	13 (/HSYNC)
4 (/VSYN) ----->	---	14 (/VSYNC)
7, 9, 11 (GND) ----->	6-9 (GND)	6-8 (GND)

### 2.1.5 EL-Display an GRIP-6A/B

GRIP-6A und GRIP-6B können in einer Sonderausführung Flachdisplays (z.b. EL-Displays SHARP) ansteuern. Der Anschluß erfolgt an Stecker N2 oder an der VG-Leiste. Der Displaybetrieb erfordert eine Sonderversion des auf der Platine eingesetzten GAL-Baustein. Für nähere Informationen setzen Sie sich bitte unter Angabe des Display-Typs mit CONITEC in Verbindung.

## 2.2 Anschluß der Tastatur

Es lassen sich verschiedene Tastaturtypen an GRIP anschließen. Der Tastaturtyp kann durch einen Host-Befehl (<ESC><ESC>'4', s. Befehlssatz) oder über das Set-Up-Menü eingestellt werden.

Beim ersten Einschalten bestimmt Jumper J7b, welche Tastatur aktiv ist:

J7a			J7b		
○	○	○	○8	○10	○12
○	○	○	○7	○9	○11

J7b offen: keine Tastatur bzw. Tastatur über Set-Up-Menü

Pos. 8-10: Tastatur = seriell (TTL/V24), ASCII, 1200 Baud, 8 Bit

Pos. 9-11: Tastatur = seriell (TTL/V24), ASCII, 600 Baud, 7 Bit

Pos. 10-12: Tastatur = PC/AT-Tastatur

Pos. 7-9: --

### 2.2.1 Serielle TTL-Tastatur

Anschluß über Leitung SKBD (N3, Stift 27). Jumper J11 muß offen sein. Die Anfangs-Baudrate wird über J7b eingestellt (600 oder 1200 Baud), andere Baudraten lassen sich vom Host aus (<ESC><ESC>'3') oder über das Set-Up-Menü einstellen.

Alternativ läßt sich die Tastatur auch über den ECB-Stecker, Stift a21, anschließen. Hierzu muß J11 in Stellung 1-2 stehen.

### 2.2.2 Serielle V24-Tastatur

Anschluß über Leitung RX1 (N3, Stift 30). Jumper J11 muß in Stellung 2-3 stehen. Die Anfangs-Baudrate wird über J7b eingestellt (600 oder

1200 Baud), andere Baudraten lassen sich vom Host aus (-> <ESC><ESC>'3') oder über das Set-Up-Menu einstellen.

### 2.2.3 PC/AT-Tastatur

Der Anschluß an den 5poligen Rundstecker erfolgt folgendermaßen:

```
3 o
 5 o (+5V) ----> +5V (N3,25)
 2 o (Data) --> KDTA (N3,17)
 4 o (GND) ----> GND (N3,16)
 1 o (Clock) ----> KCK (N3,18)
```

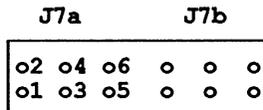
Solange dieser Tastaturtyp noch nicht im Set-Up-Menu eingetragen wurde, muß Jumper J7b in Stellung 10-12 stehen.

### 2.3 Anschluß des Hostrechners

Auch hier gibt es wieder verschiedene Möglichkeiten, GRIP an Ihren Rechner anzuschließen:

- a) über den ECB-Bus (oder an den Datenbus der CPU)
- b) über eine RS232-Schnittstelle
- c) über eine RS422/RS485-Schnittstelle
- e) über eine der Tastaturschnittstellen

Die letzte Methode - Ansteuerung über die Tastatur (LOCAL-Modus) - ist zu Testzwecken bzw. zur Eingabe der Anfangskonfiguration über das Set-Up-Menü gedacht. Die Anschlußart wird über Jumper J7a ausgewählt:



- offen: Host = ECB oder Datenbus bzw. EEPROM-Konfiguration  
 Pos. 1-3: Host = seriell (RS232/RS422), 1200 Baud  
 Pos. 2-4: Host = seriell (RS232/RS422), 9600 Baud  
 Pos. 4-6: Host = seriell (RS232/RS422), 19200 Baud  
 Pos. 3-5: Host = Tastatur (LOCAL-Modus)

#### 2.3.1 Anschluß über den ECB-Bus:

GRIP wird mit der VG-Leiste N1 in den Bus eingesteckt. Zu beachten ist, daß die mittlere b-Leiste des Steckers, die beim ECB-Bus unbelegt ist, den Grafikbus zum Anschluß eines Genlock-Interface enthält. Wenn also ein 96poliger Bus und Stecker benutzt wird, darf die b-Leiste nicht anderweitig belegt sein! Bei dem ECB-Bus (ECB-96) von CONITEC läßt sich zu diesem Zweck die b-Leiste auf jedem zweiten

Steckplatz auftrennen. Sie ist bereits auf den Plätzen 4, 6 und 8 vorgetrennt.

Außerdem sind einige weitere Signale (insbesondere bei GRIP-6B, s. Steckerbelegung) auf normalerweise freie Leitungen auf den Bus hinausgeführt. Es ist darauf zu achten, daß diese Leitungen bei Ihrem ECB-Bus nicht in Konflikt mit anderen Signalen geraten. Vergleichen Sie die Busbelegung Ihrer CPU-Platine!

Das Bus-Timing ist unkritisch; GRIP kann mit CPU-Frequenzen bis 12 MHz (Z80 oder HD64180) betrieben werden. Die Host-CPU spricht GRIP über zwei I/O-Adressen an, eine für Daten, die zweite für den Status. Die Adressen lassen sich ggfs. über einen GAL-Baustein einstellen. Gegen Aufpreis (DM 40.-) liefert CONITEC GRIP-6-Versionen mit beliebigen Bus-Adressen.

Auf der höheren Adresse (Datenport, R/W, normalerweise C1h) werden die Daten mit I/O-Befehlen eingeschrieben oder ausgelesen, auf der niedrigeren (Statusport, R/O, normalerweise C0h) der Status abgefragt. Beim Status sind nur die Bits 6 und 7 relevant.

Status:

RF	WE	xx	xx	xx	xx	xx	xx	(C0h)
D7	D6	D5	D4	D3	D2	D1	D0	(C1h)

Daten:

D0-D7: Datenleitungen zur GRIP (bidirektional)

RF: Read Buffer Full - Host kann Datenbyte auslesen

WE: Write Buffer Empty - Host kann Datenbyte einschreiben

Ist Statusbit 7 (RF) auf 1 gesetzt, so steht am Datenport ein Byte von GRIP zum Lesen bereit. Statusbit 6 (WE) signalisiert mit 1, daß das nächste Datenbyte in den Datenport geschrieben werden kann.

Zur Verdeutlichung des Ganzen folgt ein Unterprogramm in Z80-Assembler zur Ausgabe des Zeichens in CPU-Register C auf den Bildschirm:

```

GRIPOUT:  IN    A, (0COH)    ;Abfragen des Statusports
          BIT    6,A         ;Fertig zum Einschreiben?
          JR     Z,GRIPOUT   ;Wenn nicht, wieder abfragen
          LD     A,C         ;Auszugebendes Zeichen
          OUT    (0C1H),A    ;Einschreiben in den Datenport
          RET

```

Das Holen eines Datenbytes, z.b. von der Tastatur nach CPU-Register A, geht ähnlich:

```

GRIPIN:   IN    A, (0COH)    ;Status abfragen
          BIT    7,A         ;Stehen Daten von Tastatur bereit?
          JR     Z,GRIPIN    ;Wenn nicht, das Ganze noch einmal
          IN    A, (0C1H)    ;Byte holen
          RET

```

Zur Vervollständigung noch die entsprechenden Status-Routinen:

```

GRIPIST:  IN    A, (0COH)    ;Status
          BIT    7,A         ;Daten von Tastatur bereit?
          LD     A,OFFH      ;Falls ja, Return mit A=FF
          RET    NZ
          XOR    A           ;Falls nein, Return mit A=00
          RET

```

```

GRIPOST:  IN    A, (0COH)    ;Status
          BIT    6,A         ;Fertig zur Ausgabe auf Bildschirm?
          LD     A,OFFH      ;Falls ja, Return mit A=FF
          RET    NZ
          XOR    A           ;Falls nein, Return mit A=00
          RET

```

Auch an andere parallele Busse läßt sich GRIP über N1 anschließen. Zum Selektieren der Karte sind folgende Signalpegel erforderlich (bei Port-Adressen C0h,C1h):

/IORQ: LOW; A7: HIGH; A6: HIGH; A5: LOW; A4: LOW; A3:  
LOW; A2: LOW; A1: LOW.

Das Signal A0 wählt zwischen Status- (LOW) und Datenport (HIGH). An den /RD- und /WR-Leitungen sind LOW-aktive Lese- und Schreibimpulse von mindestens 300 ns Länge erforderlich. /PCL ist ein Eingang zum Rücksetzen der Karte.

/RD	/WR	A0	
0	0	x	(verboten)
0	1	0	Status lesen
0	1	1	Daten lesen
1	0	0	(inaktiv)
1	0	1	Daten schreiben
1	1	x	(inaktiv)

### 2.3.2 Anschluß über RS232-Schnittstelle

Der Anschluß erfolgt an Stecker N4, die Default-Baudraten werden über Jumper J7a eingestellt (s.o.). Das Default-Datenformat beim Einschalten ist:

8 Datenbits, 1 Stopbit, kein Paritätsbit.

Außer den Empfangs- und Sendeleitungen RX0 und TX0 verfügt die Schnittstelle über Steuerleitungen für Handshake-Betrieb (RTS0 und CTS0). Wenn diese Steuersignale nicht benutzt werden sollen, ist CTS0 mit RTS0 zu verbinden!

Die RS232-Schnittstelle besitzt zwar im Host-Betrieb einen internen Datenpuffer von 512 Byte Länge, trotzdem muß für Baudraten ab 9600 Baud oder für zeitintensive Operationen (Flächenfüllen, Vektorzeichnen, Hardcopy) RTS/CTS-Handshake oder XON/XOFF-Protokoll (-> Befehl <ESC><ESC>'3') benutzt werden.

### 2.3.3 Anschluß über RS422/RS485-Schnittstelle

Diese Schnittstelle ist optional verfügbar. Zur nachträglichen Umrüstung muß der RS232-Treiberbaustein vom Typ MAX238 (Z31) entfernt und dafür zwei RS422-Treiber/Empfänger vom Typ 75176 in die Sockel Z32 und Z33 eingesetzt werden. Zusätzlich sind im nun leeren Z31-Sockel die Pins 4 und 8 mit einer Drahtbrücke zu verbinden.

Wenn die zweite RS232-Schnittstelle für Drucker oder Tastatur (RX1/TX1) weiter verwendet werden soll, darf der MAX238-Baustein nicht entfernt werden. Stattdessen müssen die Pins 1, 2, 3, 4, 6, 7 dieses IC's abgetrennt werden.

## 2.4 Anschluß des Druckers

### 2.4.1 Paralleler Drucker an Centronics-Schnittstelle:

Anschluß an Stecker N3. Es sind nur die Signale DATA1-DATA8, /STB, BUSY und GND erforderlich. /INIT initialisiert den Drucker und wird beim Einschalten der Karte kurzzeitig aktiviert. /STB gibt einen LOW-Impuls zur Datenübernahme aus.

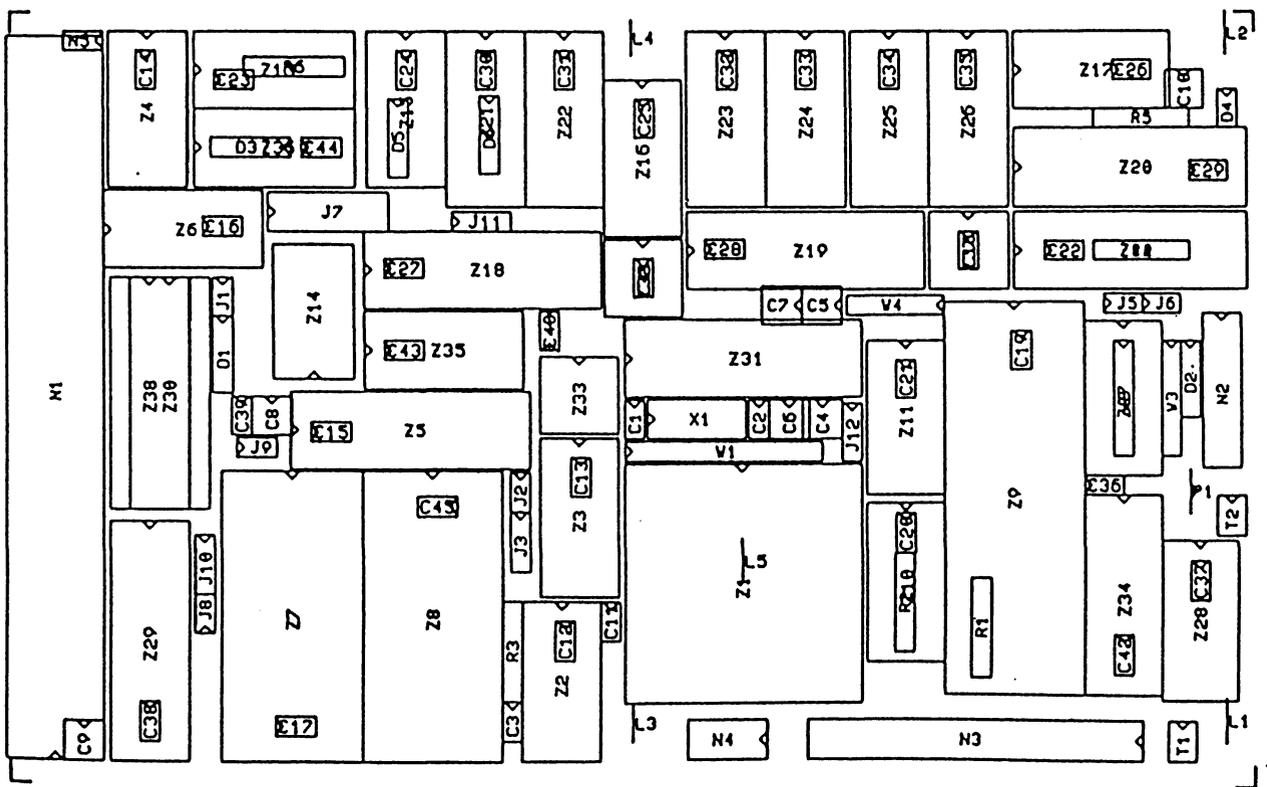
### 2.4.2 Serieller Drucker an RS232-Schnittstelle:

Anschluß an der Leitung TX1 (N3, Stift 29), Handshake über CTS1 (N3, Stift 32).

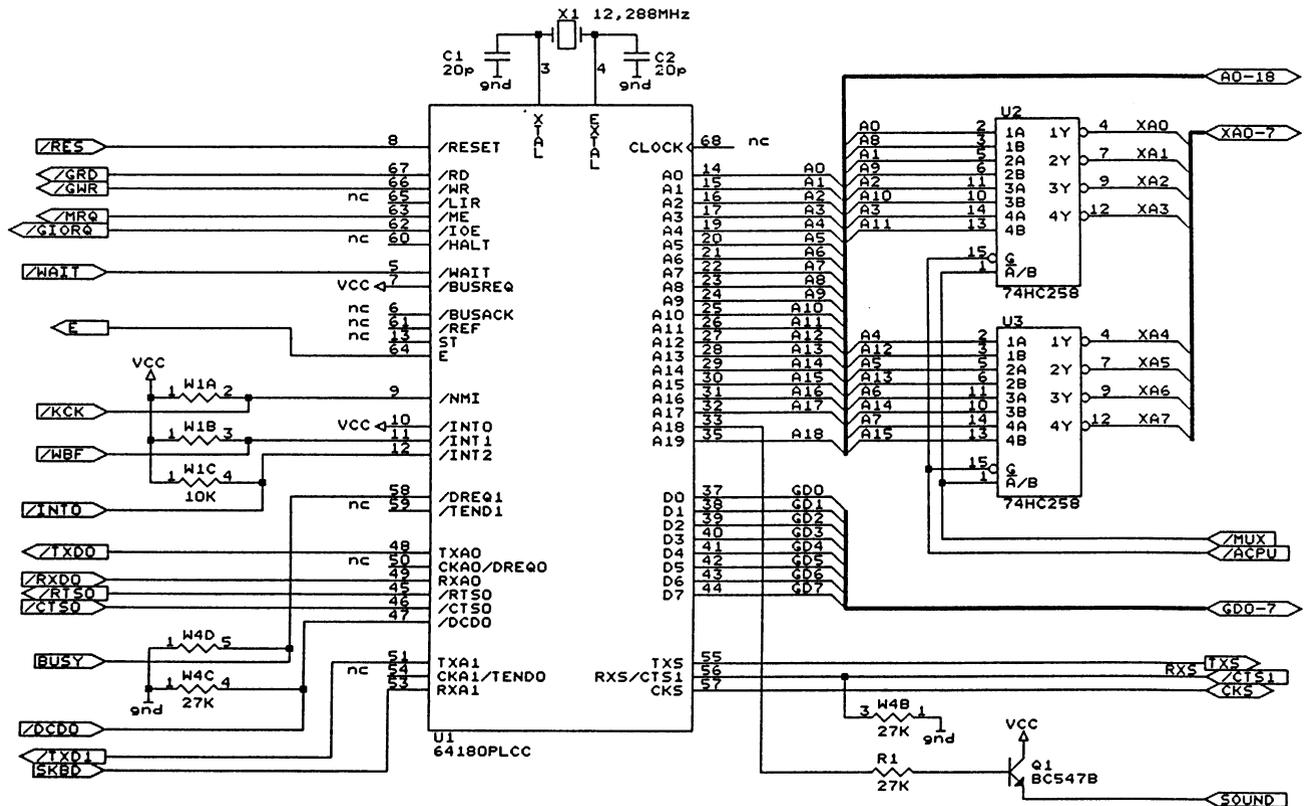
## 2.5 Anschluß eines Lautsprechers

An den SOUND-Ausgang (Stecker N2, Stift 16) kann ein hochohmiger Lautsprecher gegen GND angeschlossen werden. Das Signal kommt über einen Open-Emitter-Treiber. Die angeschlossene Impedanz sollte zwischen 100 Ohm und 50 kOhm liegen; bei Verwendung einer Piezo-Kapsel ist ein 1k-Widerstand der Kapsel parallelzuschalten.

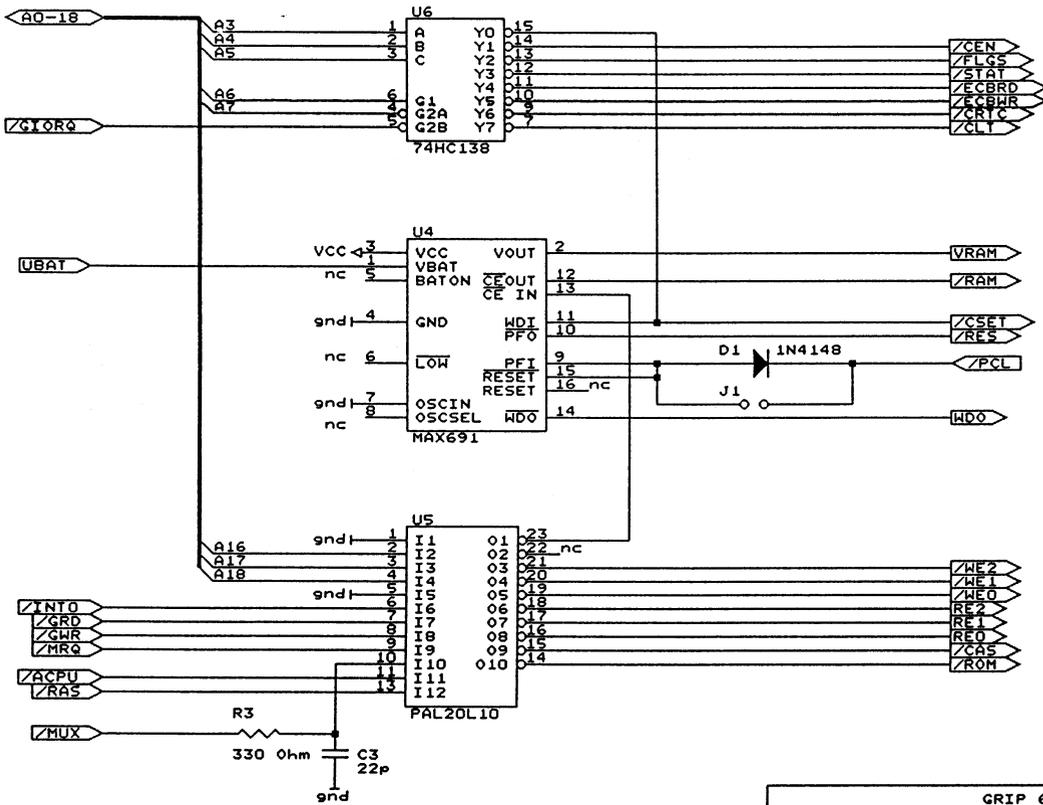
②



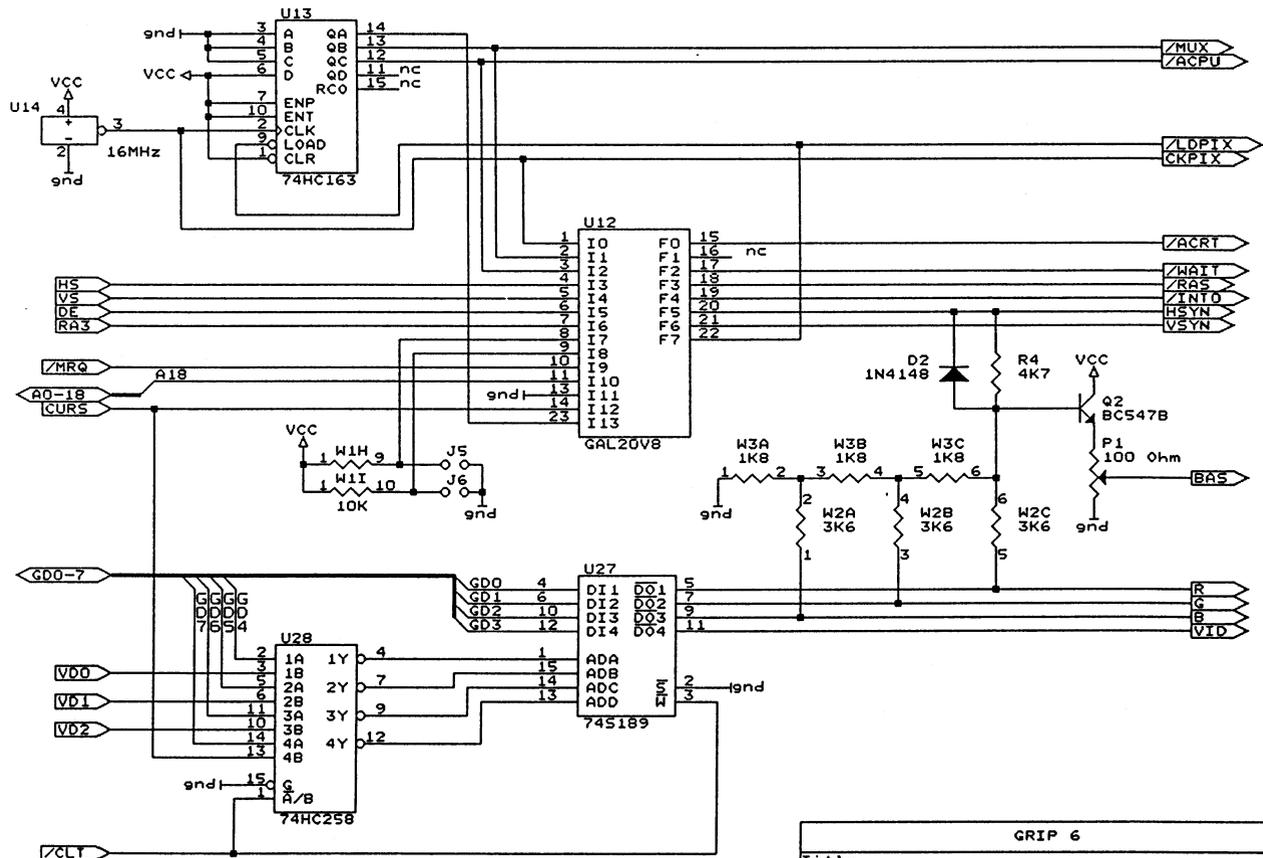
②



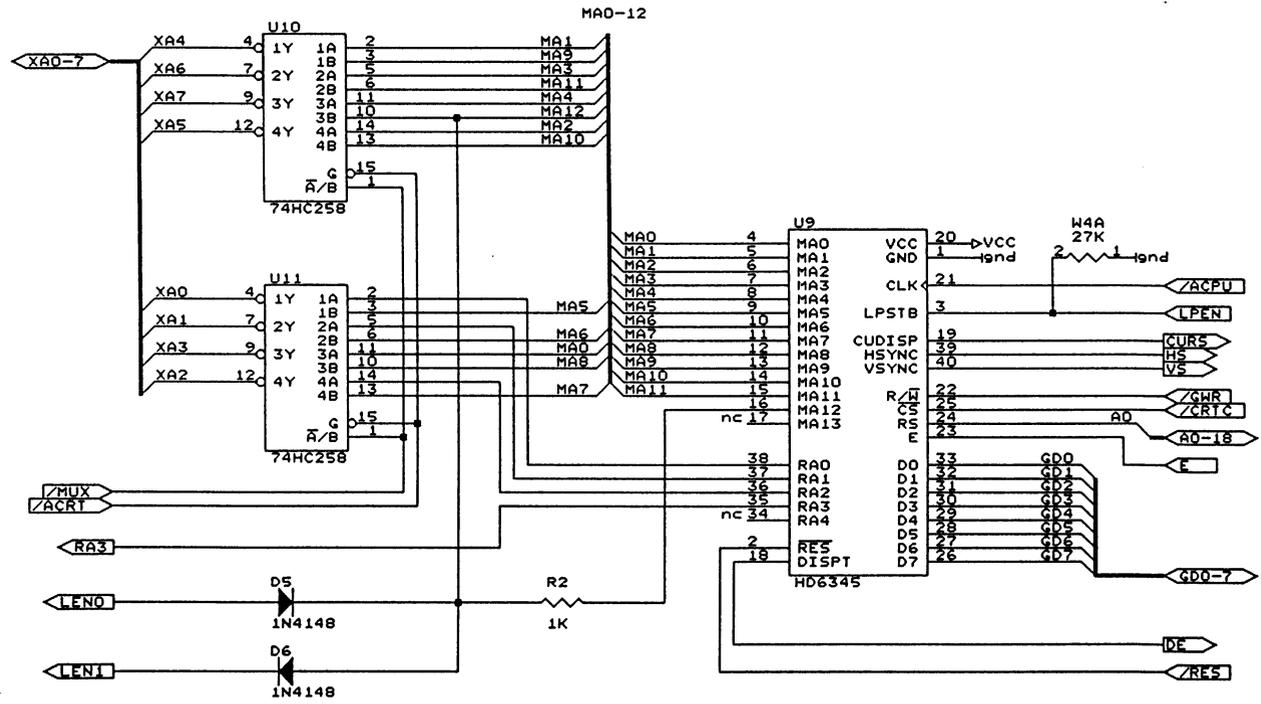
GRIP 6		
Title		
CPU		
Size Document Number		REV
A	Conitec JCL	0
Date:	August 9, 1990	Sheet 1 of 12



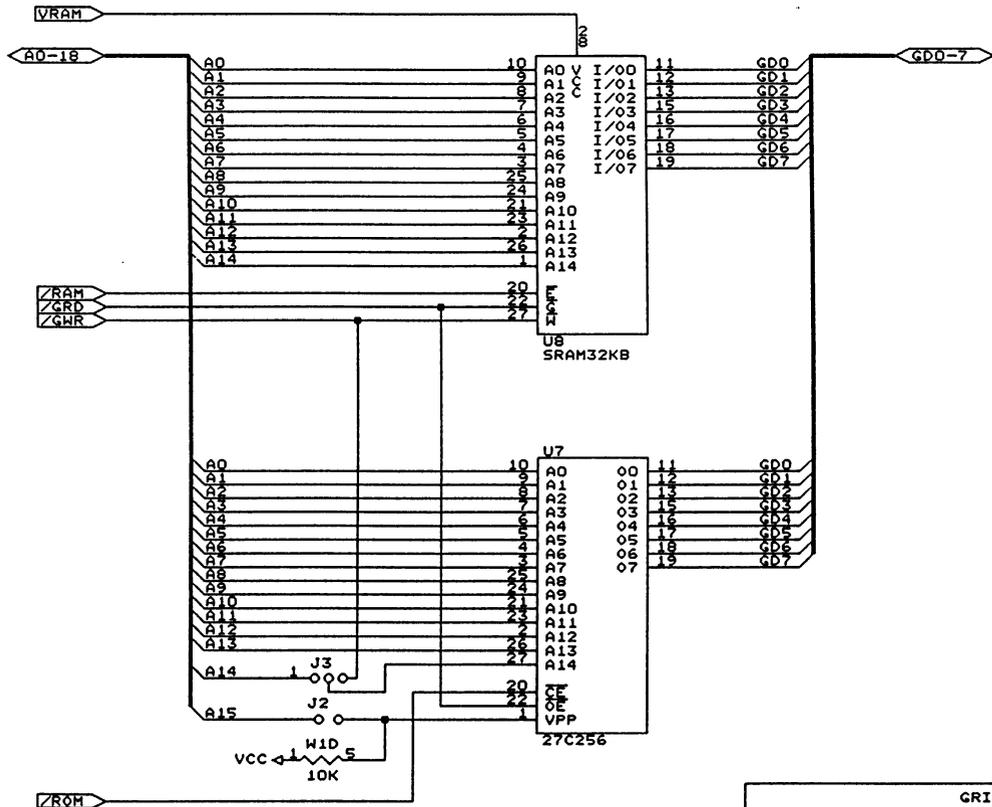
GRIP 6		
Title		
Decoder		
Size	Document Number	REV
A	Conitec JCL	0
Date:	August 9, 1990	Sheet 2 of 12



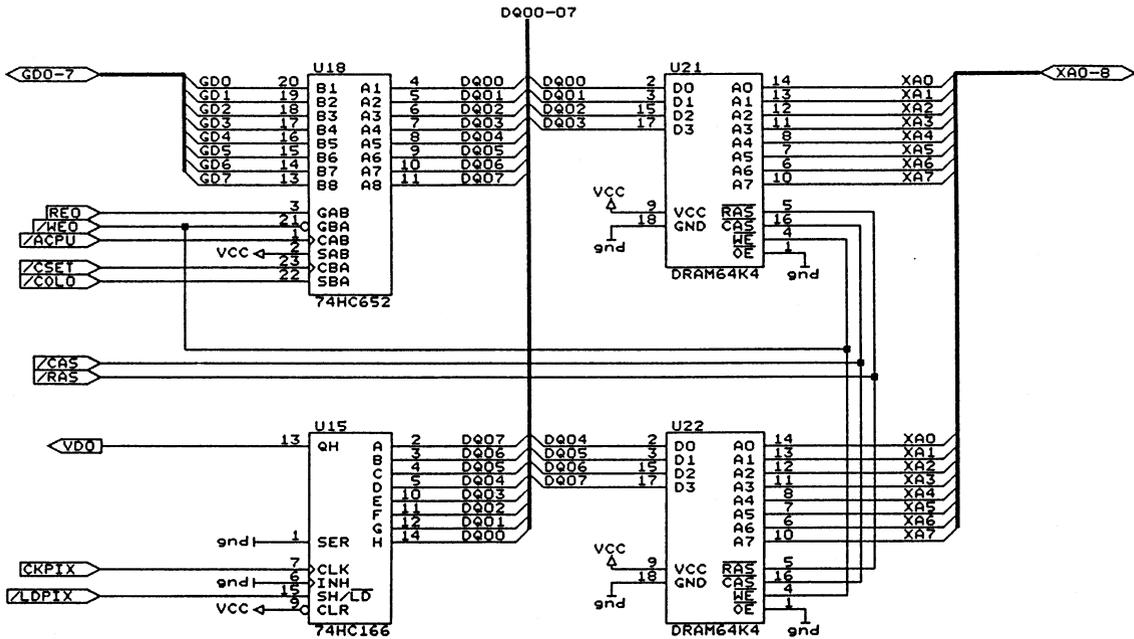
GRIP 6		
Title		
Timer/Video		
Size Document Number		
A	Conitec JCL	REV 0
Date:	August 9, 1990	Sheet 3 of 11



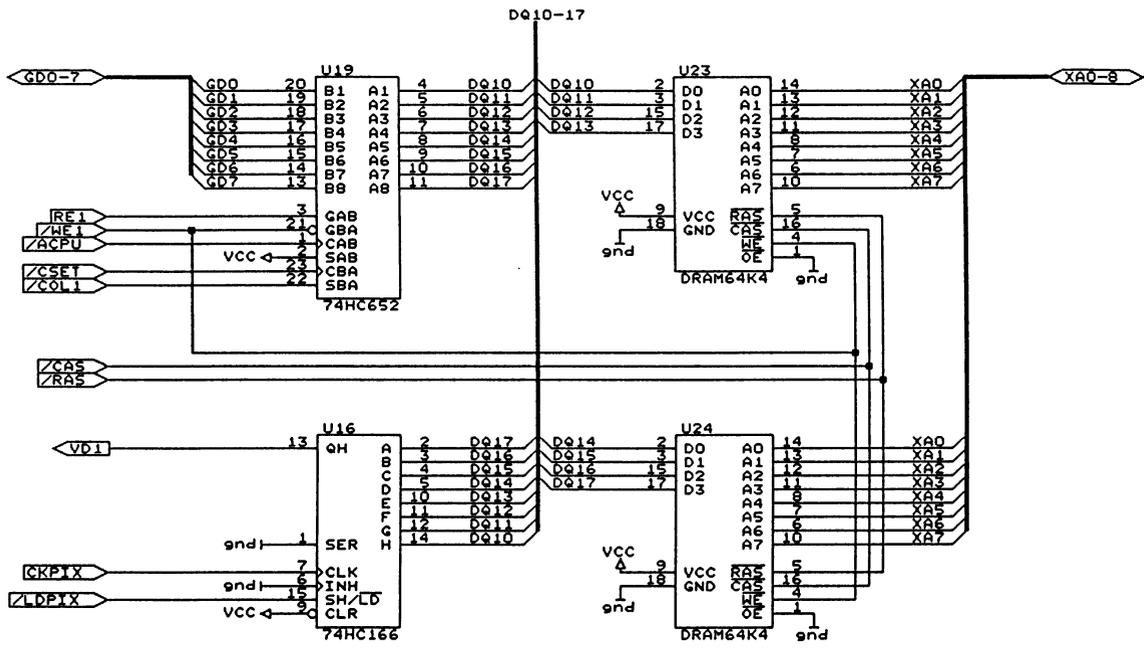
GRIP 6			
Title			
Control			
Size	Document Number		REV
A	Conitec JCL		0
Date:	August 9, 1990	Sheet	4 of 12



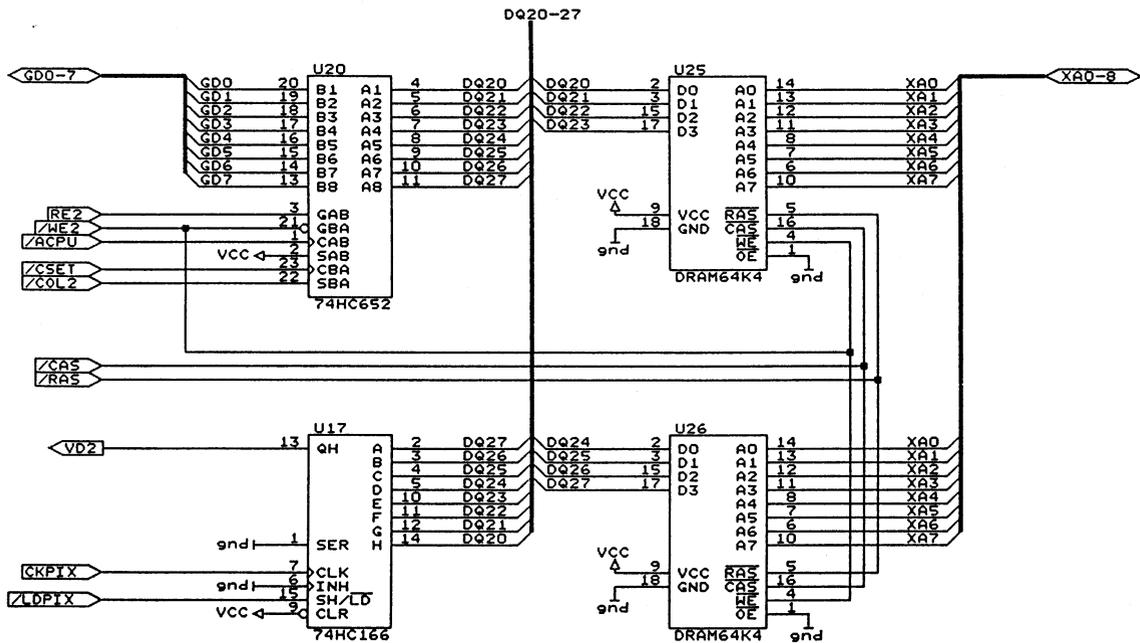
GRIP 6			
Title			
Memory			
Size	Document Number	REV	
A	Conitec JCL	0	
Date:	August 9, 1990	Sheet	5 of 12



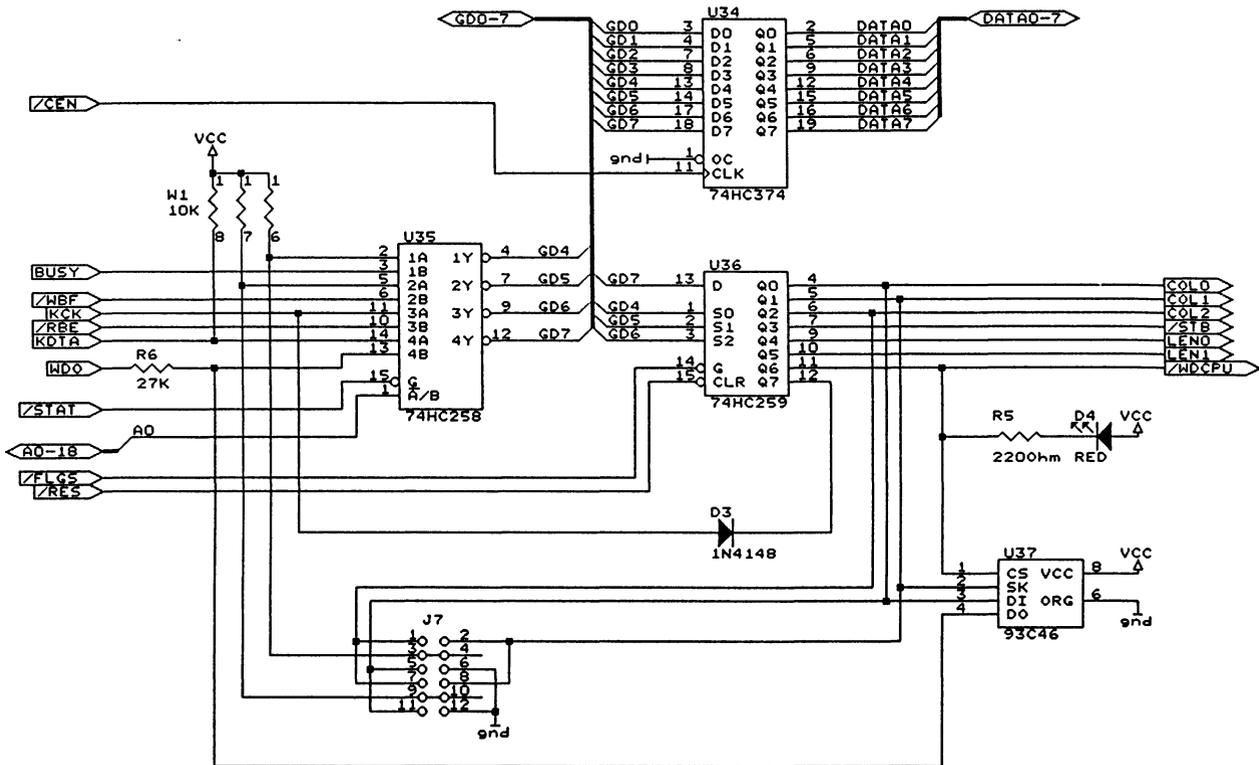
GRIP 6		
Title		
RAM		
Size Document Number		REV
A	Conitec JCL	0
Date:	August 9, 1990	Sheet 6 of 12



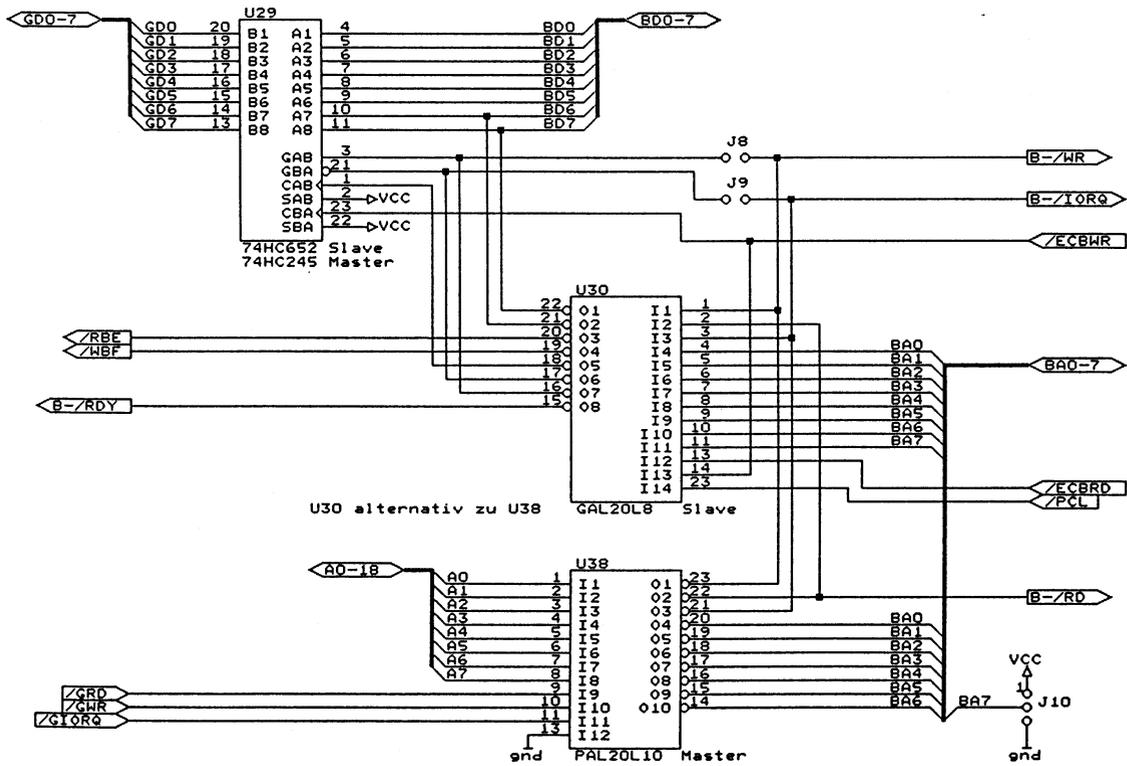
GRIP 6		
Title		
RAM 1		
Size Document Number		
A	Conitec JCL	REV 0
Date:	August 9, 1990	Sheet 7 of 12



GRIP 6	
Title	RAM 2
Size Document Number	REV
A	Conitec JCL
Date: August 9, 1990	Sheet 8 of 12

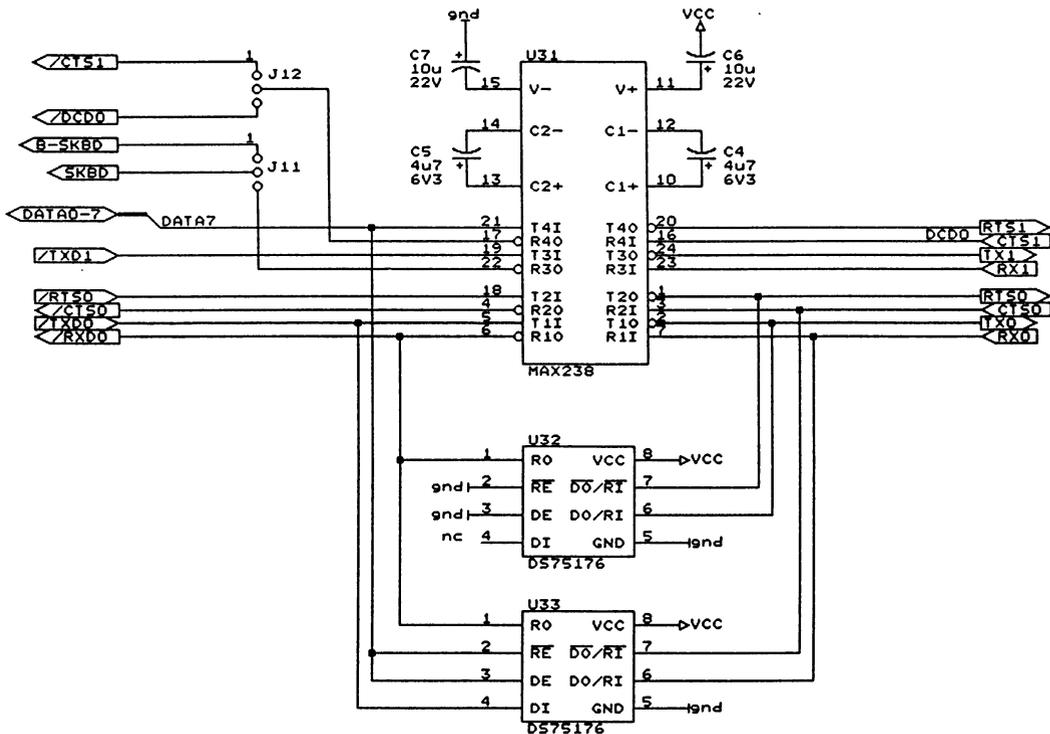


GRIP 6			
Title		Ports	
Size	Document Number	REV	
A	Conitec JCL	0	
Date:	August 10, 1990	Sheet	9 of 12



U30 alternativ zu U38

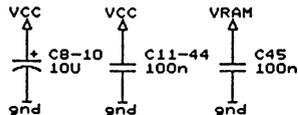
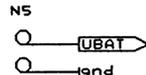
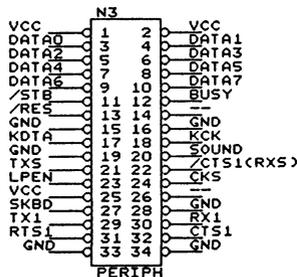
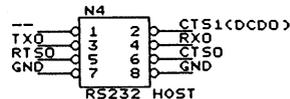
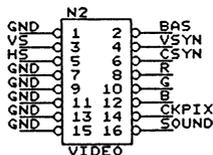
GRIP 6		
Title		
ECB		
Size	Document Number	REV
A	Conitec JCL	0
Date:	August 10, 1990	Sheet 10 of 12



GRIP 6		
Title		
Ser.Buffer		
Size Document Number		REV
A	Conitec JCL	0
Date:	August 10, 1990	Sheet 11 of 12

N1 ECB-Bus-Stecker

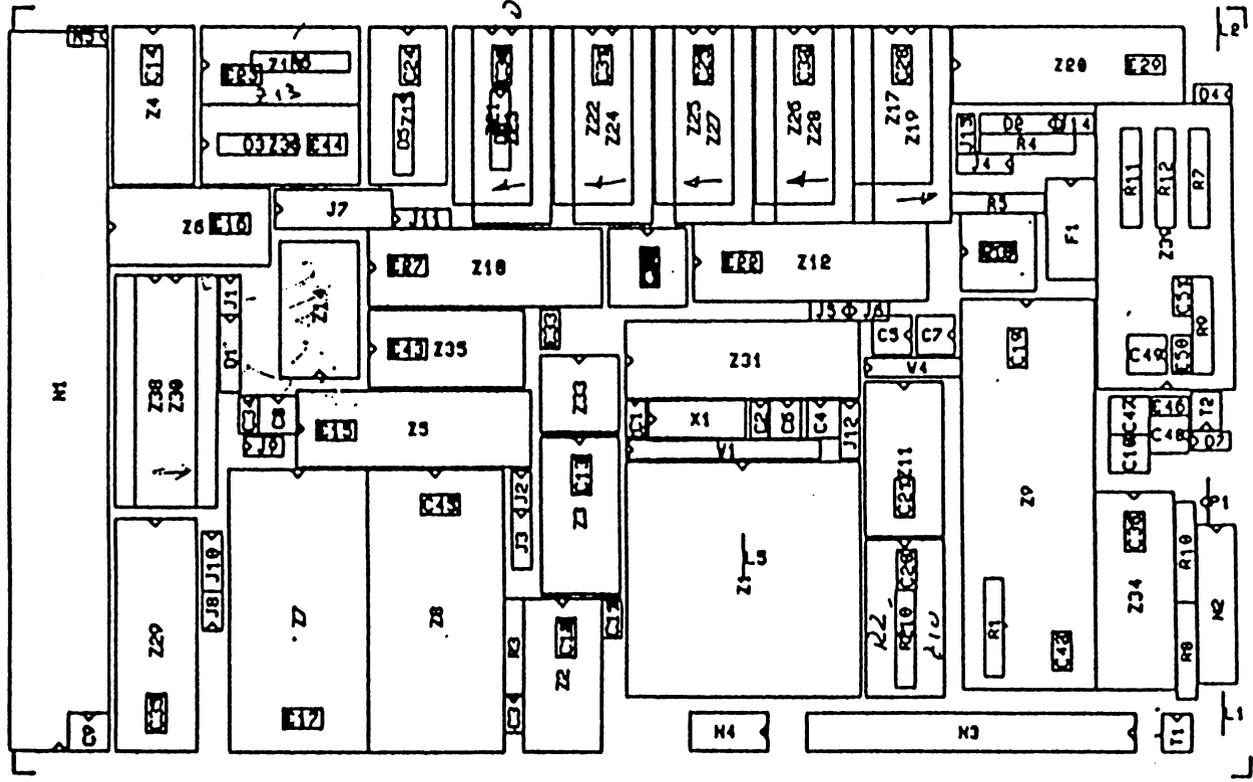
	a	b	c
01	VCC	VCC	VCC
02	B-D5	--	B-D0
03	B-D6	/GRD	B-D7
04	B-D3	/GHR	B-D2
05	B-D4	GD0	B-A0
06	B-A2	GD1	B-A3
07	B-A4	XA6	B-A1
08	B-A5	XA4	--
09	B-A6	GD2	B-A7
10	--	GD3	--
11	--	XA0	IE1
12	--	XA1	---
13	--	XA8	---
14	--	GD4	B-D1
15	TX0	GD5	DATA7
16	RTS0	XA5	IE0
17	--	XA7	---
18	--	GD6	---
19	RXD	GD7	---
20	--	/CAS	---
21	B-SKBD	XA3	---
22	CTS0	XA2	B-/WR
23	--	/INT0	/WDGPU
24	UBAT	/GIORQ	B-/RD
25	--	/RAS	---
26	B-/RDY	VID	/PCL
27	B-/IORQ	VSYN	---
28	--	HSYN	---
29	--	/ACPU	---
30	--	/LDPIX	---
31	--	CKPIX	---
32	GND	GND	GND

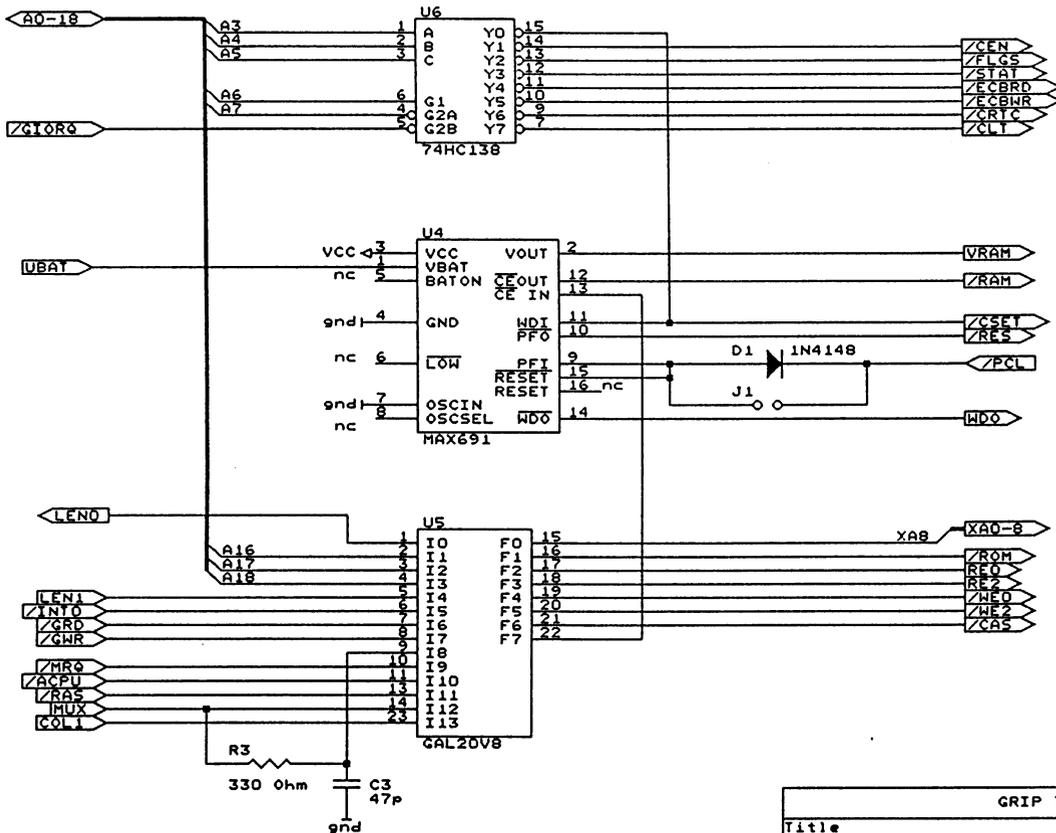


GRIP 6			
Title			
Plug			
Size	Document Number	REV	
A	Conitec JCL	0	
Date:	August 10, 1990	Sheet	12 of 12

26

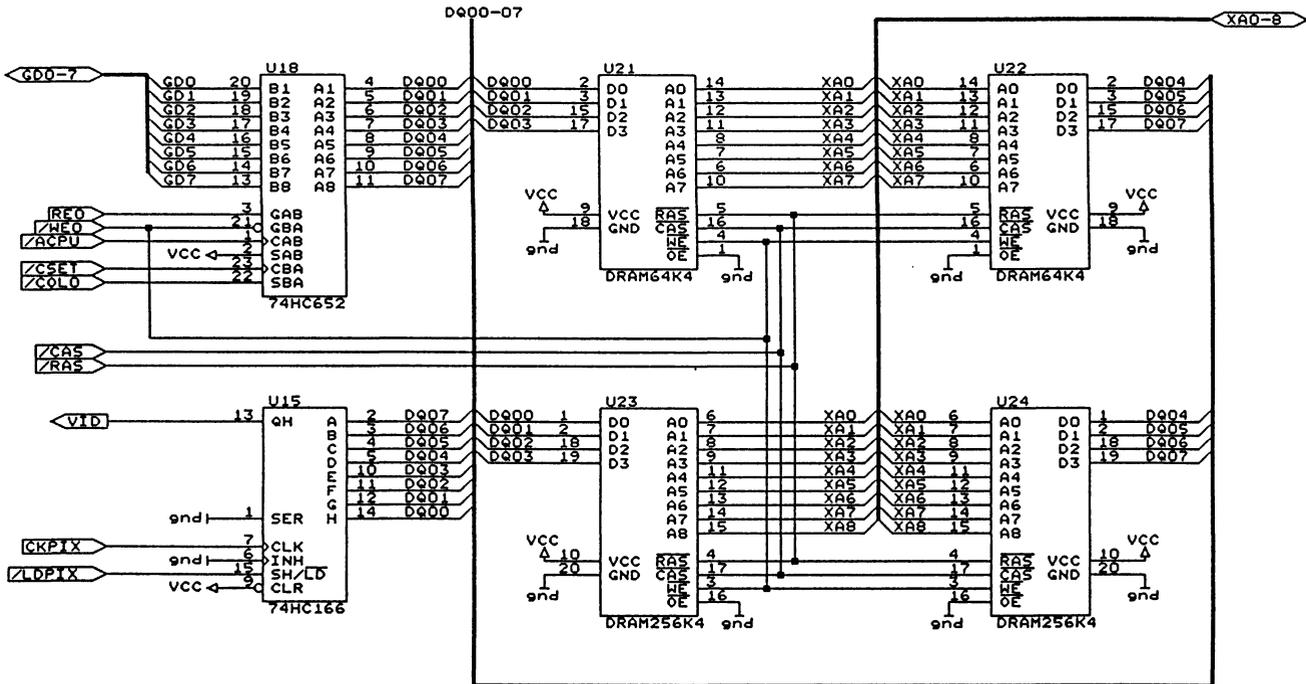
26





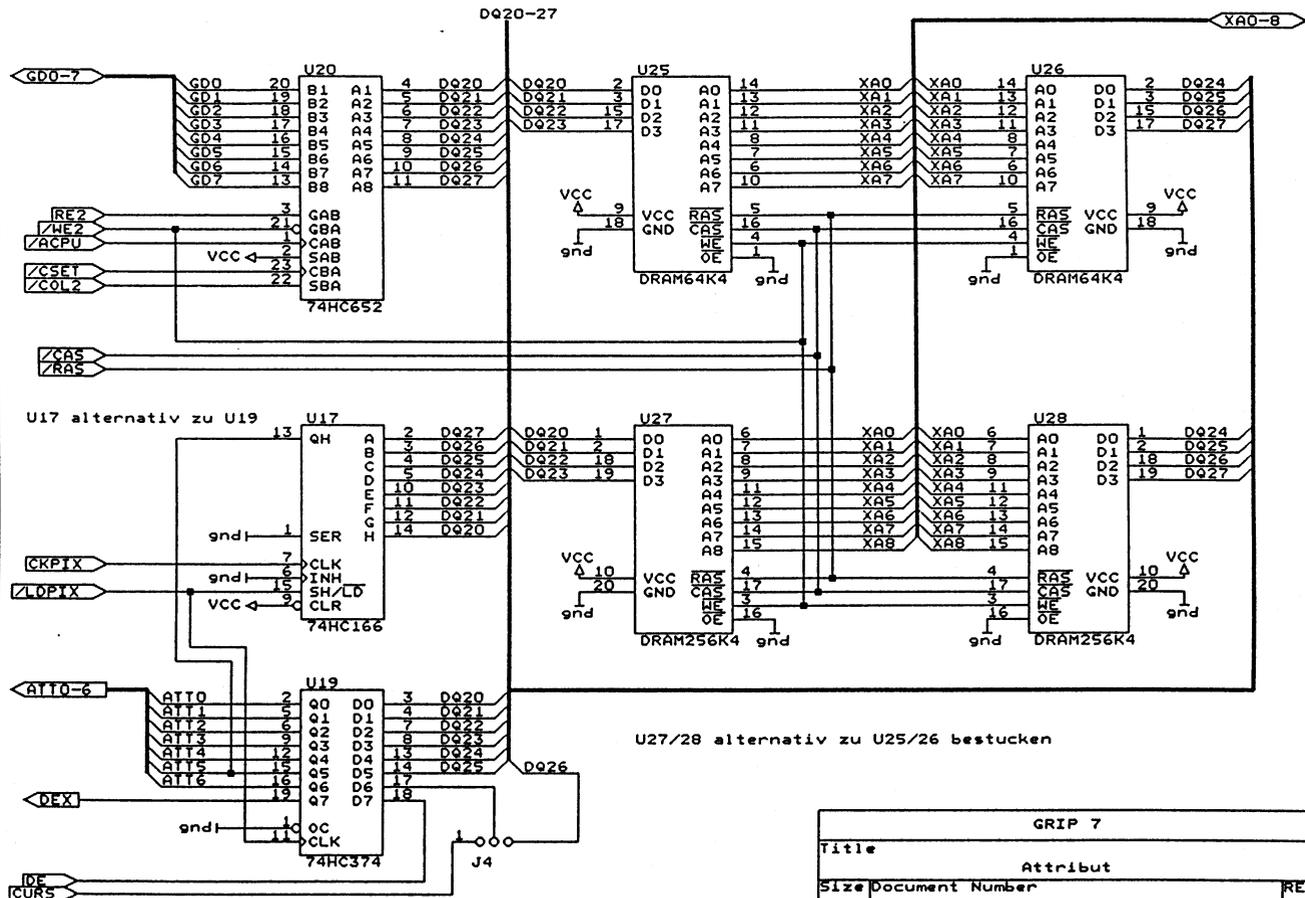
GRIP 7		
Title		
Decoder		
Size Document Number		REV
A	Conitec JCL	0
Date:	July 30, 1990	Sheet 2 of 11





U23/24 alternativ zu U21/22 bestücken

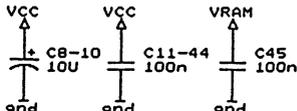
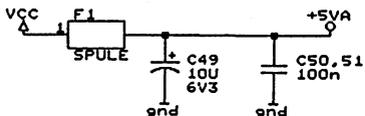
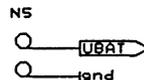
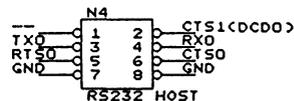
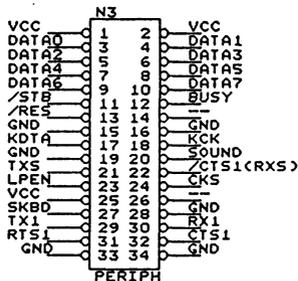
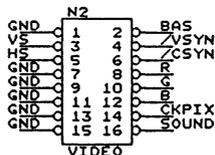
GRIP 7			
Title	RAM		
Size	Document Number		REV
A	Conitec JCL		0
Date:	August 9, 1990	Sheet	6 of 11



GRIP 7	
Title	Attribut
Size Document Number	REV
A	0
Conitec JCL	
Date: August 9, 1990	Sheet 7 of 11

# N1 ECB-Bus-Stecker

	a	b	c
01	VCC	VCC	VCC
02	B-D5	---	B-D0
03	B-D6	/GRD	B-D7
04	B-D3	/GHR	B-D2
05	B-D4	GD0	B-A0
06	B-A2	GD1	B-A3
07	B-A4	XA6	B-A1
08	B-A5	XA4	---
09	B-A6	GD2	B-A7
10	---	GD3	---
11	---	XA0	IE1
12	---	XA1	---
13	---	XA8	---
14	---	GD4	B-D1
15	---	GD5	---
16	---	XA5	IE0
17	---	XA7	---
18	---	GD6	---
19	---	GD7	---
20	---	/CAS	---
21	B-SKBD	XA3	---
22	---	XA2	B-/HR
23	---	/INT0	---
24	UBAT	/GIORQ	B-/RD
25	---	/RAS	---
26	B-/RDY	VID	/PCL
27	B-/IORQ	/VSY	---
28	---	/CSYN	---
29	---	/ACPU	---
30	---	/LDPIX	---
31	---	CKPIX	---
32	GND	GND	GND



GRIP 7		
Title	Plug / Circuit	
Size	Document Number	REV
A	Conitec JCL	0
Date:	July 30, 1990	Sheet 11 of 11