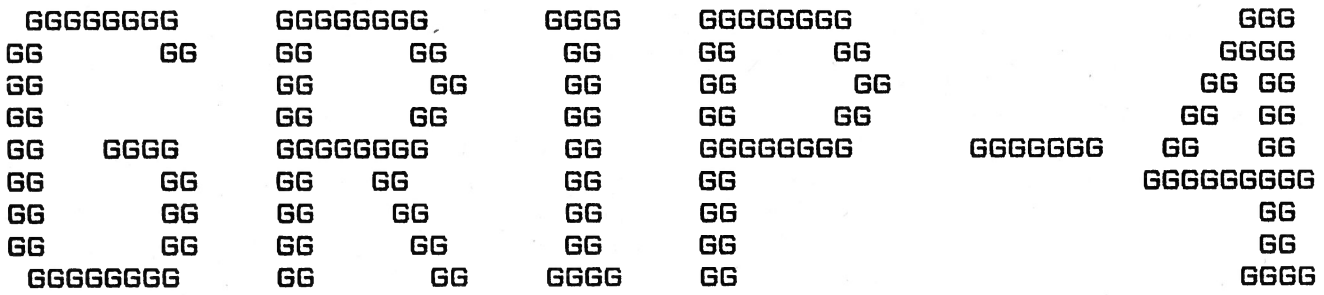


Grafik-Interface-Prozessor

Handbuch

CONITEC DATENSYSTEME

Postfach 11 06 22 · D-6100 Darmstadt 11
Telefon (0 61 51) 2 60 13 · Telex 4 197 298

GRIP-4

Grafik-I/O-Prozessor

Inhaltsverzeichnis

GRIP-4 - t e c h n i s c h e D a t e n

- **Hardware**: eigene Z80-CPU; 32 KB EPROM; 72 KB RAM; 4 Timer; 16 Interrupt-Kanäle; 2 serielle, 3 parallele Schnittstellen.

- **Grafik**: Teilemulation des **Tektronix-4010**-Terminals; Auflösung bis **768x560 Bildpunkte**, erweiterbar auf Farbgrafik (max. 128 aus 4096 Farben); Grafik gleichzeitig mit Text darstellbar; leistungsfähige **Vektorbefehle** (Zeichnen/Löschen/Invertieren in 5 Stricharten); Punkt- und Inkremental-Modus; **Hardcopy-Funktion** auf grafikfähigen Matrixdrucker.

- **Text**: Teilemulation des **TVI950**-Terminals; variables Textformat, z.b. 40x24, 80x24 oder 96x35; variable Zeichenmatrix 8x8 bis 16x16; **9 Zeichensätze** in zwei Schriftgrößen; Formelsymbole; Indizes; Strichgraphik; alle Zeichen im RAM frei umdefinierbar; voll WORDSTAR-/MULTIPLAN-tauglich; Textaufbau mit **5000 Zeichen/sec**; 7 kombinierbare **Attribute**: 2. Zeichensatz, hoch/tiefgestellt, unsichtbar, Breitschrift, Invertieren, Durchstreichen und Unterstreichen.

- **Features**: Umschaltbare System- und User-Statuszeile; **Echtzeituhr**; flimmerfreies Scrollen in alle Richtungen; Monitor-Anpassung per Software; Testbild; frei definierbarer Cursor; Glockensignal (BELL); Download für Userprogramm; direkter Zugriff auf's Video-RAM und auf alle Schnittstellen.

- **Host-Schnittstelle**: **V24/RS232** oder **ECB-Bus**; bidirektionaler Transfer, 7 oder 8 Bit; RTS/CTS-Handshake oder XON/XOFF-Protokoll; Baudraten 50 - 9600 Baud; ECB-Bus-Anschluß mit Daten- und Statusport auf zwei I/O-Adressen; benötigt werden nur D0-7, A0-7, /RD, /WR, /IORQ, darum auch für andere Busnormen und Prozessoren einsetzbar.

- **Peripherie-Schnittstellen**: **V24/RS232-Vollduplex-Schnittstelle** (bei ECB-Bus-Betrieb) und zusätzliche **Centronics-Schnittstelle**; Datenformat und interne Baudraten (50 - 9600 Baud) für Sender und Empfänger vom Host aus einstellbar; Eingang für externen Baudratentakt; abschaltbarer **30-KByte-Druckerspooler**.

- **Tastatur-Schnittstelle**: serieller und paralleler Tastatur-Anschluß, 7 oder 8 Bit; Baudrate per Software einstellbar (max. 4800 Baud); Tastenklick mit einstellbarer Lautstärke; programmierbare **Umcode-Tabelle**; eigener Tastaturpuffer mit Sichtfeld in der Statuszeile.

- **Lichtgriffeleingang** mit einstellbarer Ansprechschwelle.

- **Soundgenerator**: beliebige Tonfolgen programmierbar; Frequenzbereich ca. 40 Hz - 60 kHz; 3 Lautstärkestufen; Speicher für 512 Töne; Ausgang für NF-Endstufe oder hochohmigen Lautsprecher.

- **Monitoranschluß**: BAS-Signal; Video- und positive/negative Sync-Ausgänge; Bildfrequenz 50 Hz, Zeilenfrequenz 15.625 kHz; Eingang für externen Bildpunkttakt (Fremdsynchronisation).

- **Grafikbus** zum Anschluß von Farbzusatz (GRIP-COLOR).

- Stromversorgung: 5V (ca. 900 mA) und +/-12V (ca. 20 mA).

- Platinengröße: 100x160 mm (Standard-Europaformat).

I. --- Einführung ---

I.1: Die Grundkarte GRIP-4

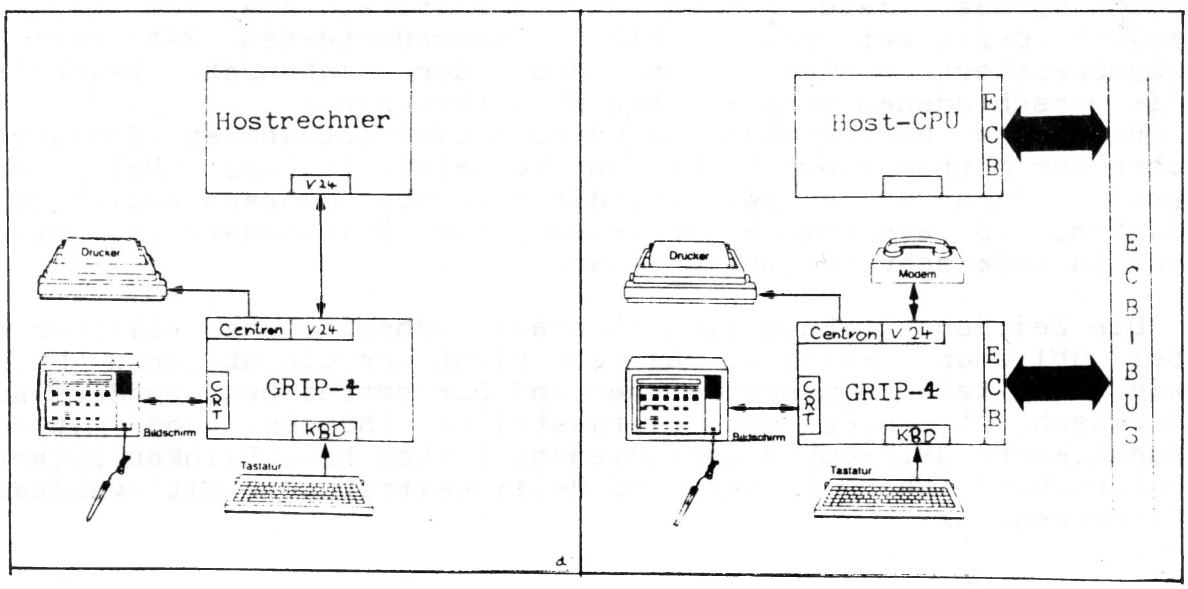
Die Multifunktionskarte GRIP-4 (Grafik-I/O-Prozessor) vereinigt alle Ein/Ausgabekanäle eines Rechnersystems auf einer einzigen Europakarte. Sie läßt sich ähnlich wie ein Terminal ansteuern und enthält ein Videointerface für Text und hochauflösende Grafik, mehrere Schnittstellen zur Verbindung von Hostrechner und Peripherie sowie Anschlüsse für Lichtgriffel und Lautsprecher.

Ihre Stärke sind die vielen implementierten Funktionen, die ein ganzes Bündel von Zusatzkarten und -Software ersetzen. GRIP besitzt dazu einen eigenen Z80-Prozessor mit einem 32-KByte-Betriebssystem.

GRIP läßt sich auf zwei verschiedene Arten vom Host-Rechner ansteuern:

- Über eine serielle **V24/RS232-Schnittstelle**, ähnlich wie ein konventionelles Terminal,

- oder direkt über den Datenbus des Host-Prozessors; hierzu besitzt die Karte einen **ECB-Bus-Anschluß**. Beim Betrieb zusammen mit der Prozessorkarte PROF wird z.B. von dieser letzten Möglichkeit Gebrauch gemacht. Wegen der unkritischen Ansteuerung lassen sich über eine Adapterkarte jedoch auch andere Busnormen anschließen.



In beiden Fällen wirkt die Karte wie ein Datenkonzentrator; der Hostrechner kommuniziert mit ihr über eine einzige Schnittstelle (bzw. zwei I/O-Ports) und kann auf diese Weise viele verschiedene (langsamere) Peripherieeinheiten bedienen. Alle CP/M-Devices (CON:,LST:,AUX:) werden durch GRIP abgedeckt. Wegen der Spoolerfunktion der GRIP kann dabei die Übertragungsgeschwindigkeit wesentlich höher sein als bei direktem Anschluß.

Die einzelnen Peripheriefunktionen der GRIP sind im folgenden kurz näher erläutert:

A. Zunächst läßt sich die Karte mit angeschlossenem Bildschirm und Tastatur wie ein normales **Text-Terminal** einsetzen. Sie emuliert das **Televideo-950-Standardterminal** und kann so auf einfache Weise an die üblichen Softwarepakete, wie z.b WORDSTAR, CALCSTAR, DBASE, MULTIPLAN usw. angepaßt werden. Für die 'Spreadsheet'-Programme ist auch eine **Strichgrafik** implementiert, die z.b. von MULTIPLAN unterstützt wird.

Durch das Hardware-Konzept mit einem volltransparenten, synchronen Video-RAM ist das Bild auch bei Funktionen wie Scrollen, Zeichenverschieben usw. absolut ruhig und stabil. Flackern oder Dunkelschalten, wie bei anderen Video-Karten (auch bei den Vorgängern GRIP-2 und -3), gibt es nicht. Zeichenübertragung und Bildaufbau sind extrem schnell (ca. 30.000 Baud!), das Scrollen erfolgt per Hardware und benötigt darum keine Wartezeit.

Natürlich kann GRIP im Text-Modus noch einiges mehr als das normale Televideo-Terminal. Die Zeichen lassen sich in verschiedenen Größen (8x8..16x16 Bildpunkte) und mit einstellbarem Textformat (40x24..96x35 Zeilen) darstellen.

GRIP enthält 6 verschiedene Zeichensätze mit unterschiedlicher Zeichengröße; dazu kommen noch 2 weitere, die vom Benutzer **selbst definiert** und in GRIP's akkugepuffertem RAM resident abgespeichert werden können. Jeder der Zeichensätze beinhaltet die verschiedenen europäischen Sonderzeichen.
Besonders hervorzuheben sind noch die verfügbaren Sonderzeichen für mathematisch/technische Formeldarstellung. Neben Integral-, Wurzel- u.a. Zeichen, die auch über mehrere Zeilen gehen dürfen, ist ein kompletter griechischer Zeichensatz in Normal- und in Indexschrift implementiert.

Die **Zeichenattribute** sind ebenfalls erheblich vielfältiger als bei üblichen Terminals. Es gibt nicht nur die üblichen Sachen wie Invertieren, Unterstreichen und Durchstreichen, sondern auch Breitschrift, hoch- und tiefgestellte Indizes, und mit der Zusatzkarte **GRIP-COLOR** verschiedene Farben bzw. Blinken zwischen unterschiedlichen Farben und Helligkeitsstufen mit wählbarer Blinkfrequenz.

In der **System-Statuszeile** am oberen Bildrand wird der Betriebszustand der Karte mit Symbolen und Kennziffern angezeigt. Außerdem enthält diese Statuszeile noch zwei Besonderheiten:

- eine **Echtzeituhr**, die beim Einschalten gestellt werden kann und dann ständig die aktuelle Zeit anzeigt,
- und ein **Tastaturfenster**, das einen Blick auf die letzten 15 Bytes im Tastaturpuffer (s.u.) erlaubt.

Für Kommando-Menues o.ä. gibt es eine zweite User-Statuszeile am unteren Bildrand, die vom Benutzer beschrieben werden kann.

Für Sonderzwecke sind weitere Features eingebaut: Scrollen in alle Richtungen - links,rechts,oben,unten -, ein frei definierbarer Cursor, wählbarer Zeilenabstand, ein Testbild zum Monitor-Einstellen und noch einiges mehr.

B. Die interessanteste Funktion von GRIP ist natürlich die **hochauflösende Grafik**. Sie verfügt über 768x560 Bildpunkte und wird durch leistungsfähige Kommandos unterstützt (schnelles Vektorzeichnen in verschiedenen Linienarten, ab Mitte 1986 auch Kreisfunktionen und Flächenfüllen). Die Standard-Software emuliert ein Tektronix-Grafikterminal. Neben dem Vektormodus sind noch Alphamodus, Punktmodus und ein plotterähnlicher Inkrementalmodus implementiert. Für besondere Anwendungen sind Software-Pakete wie z.b. eine **3D-Grafikbibliothek** für TURBO-PASCAL erhältlich.

Farbgrafik (16 oder 128 aus einer Palette von 4096 frei definierbare Farben bzw. Graustufen pro Bildpunkt) ist mit der Zusatzkarte **GRIP-COLOR** möglich. Mit Hilfe dieser Zusatzkarte läßt sich auch ein zweiter Bildschirm anschließen, so daß an der GRIP/COLOR-Kombination ein Text- und ein Grafikterminal gleichzeitig nebeneinander betrieben werden können.

Bei reduzierter Auflösung (768x280) wird eine zweite Bildschirmseite frei, in der die Grafik 'im Hintergrund' aufgebaut werden kann, während noch das vorherige Bild angezeigt wird. Auf diese Weise lassen sich rasche Bildwechsel durchführen, ohne daß man den Bildaufbau sieht.

Der Hostrechner kann auch direkt auf GRIP's Grafik-Speicher zugreifen. So lassen sich fertige Bilder z.b. von Diskette laden; die Übertragung eines kompletten Schwarzweißbildes über den ECB-Bus dauert dabei etwa zwei Sekunden.

Damit man auch etwas schwarz auf weiß in der Hand hat, gibt es neben dem Abspeichern auf Diskette die Möglichkeit, fertige Bilder in voller Auflösung auf einen Drucker auszugeben. Der **HARD-COPY**-Befehl unterstützt alle üblichen Farb- oder Schwarzweiß-Matrixdrucker (z.b. EPSON), die spaltenweise Punktgrafik ausgeben können.

C. Zur Dateneingabe dienen drei verschiedene Tastatur-Eingänge: Es können parallele, serielle (V24 oder TTL) oder IBM-ähnliche ASCII-Tastaturen angeschlossen werden, sogar - wenn man das will - alle drei gleichzeitig. Ein per Software zuschaltbarer 'Tastenklick' läßt eine akustische Rückmeldung für jeden Tastendruck ertönen.

Über eine 1000 Zeichen große **Umcode-Tabelle** lassen sich beliebige Tasten mit Zeichenfolgen belegen, die dann anstelle des Tastencodes zum Hostrechner gesendet werden. Das erhöht in vielen Fällen den Bedienungskomfort ("Funktionstasten"). Natürlich kann die Umcodetabelle jederzeit vom Hostrechner umprogrammiert werden, so daß man sich für jedes Programm eine eigene Tastenbelegung definieren kann.

Die letzten 64 Eingaben werden im **Tastaturpuffer** zwischengespeichert, die letzten 15 in der Statuszeile angezeigt. So kann man bereits Befehle eintippen, während der Rechner z.b. noch ein Programm lädt und die Tastatur nicht abfragt. Wer häufig mit diskettenzugriffsintensiven Programmen arbeitet, wird dies zu schätzen wissen.

D. Insbesondere für Grafik sind alternative Eingabemöglichkeiten von Vorteil. Neben der Möglichkeit, am seriellen Tastatureingang oder an der V24-Schnittstelle ein **Grafiktablett** (z.b. PREH Bitmap) oder eine **Maus** anzuschließen, besitzt GRIP noch einen **Lichtgriffeleingang**. Der Lichtgriffel ist sicherlich die preiswerteste und wohl auch die direkteste Methode zur Eingabe am Bildschirm. GRIP kann den Lichtgriffelstandort mit einer Genauigkeit von 4x2 Pixels (etwa 1 mm_2 am Bildschirm) orten.

E. Wenn GRIP am ECB-Bus liegt, wird die **V24-Schnittstelle** frei zum Anschluß irgendeiner Peripherieeinheit, z.b. eines Modems, an den Host-Rechner. Der 'AUX:'-Device von CP/M kann dabei zur Bedienung implementiert werden. Baudraten, Datenformat und Protokolle (RTS/CTS oder XON/XOFF) sind frei programmierbar.

F. Natürlich braucht jeder Rechner ein **Druckerinterface**, für das auf der GRIP wahlweise die CENTRONICS- und die V24-Schnittstelle zur Verfügung stehen. Das Interface kann auf einfache Weise z.b. über den 'LST:'-Device von CP/M bedient werden. Die HARDCOPY-Funktion wirkt ebenfalls auf dieses Interface.

Für die Datenübertragung zum Drucker läßt sich ein Zwischenspeicher (**Spooler**) von 32 KByte Länge zuschalten. Damit können ca. 10 bis 15 DIN-A4-Seiten Text auf einmal an die Karte ausgegeben und dann mit größtmöglicher Geschwindigkeit gedruckt werden, während der Host-Prozessor bereits wieder für andere Aufgaben frei ist.

Der Spooler-Betrieb ist immer dann sinnvoll, wenn ein schnelles System Daten an ein langsames Peripheriegerät sendet, wie z.b. an einen Drucker oder Plotter. Er läuft auf der GRIP im Hintergrund ab, so daß man während des Ausdruckens normal am Bildschirm weiterarbeiten kann.

G. Last not least besitzt GRIP einen **Soundgenerator**, z.b. um Schlachtenlärm für Computerspiele zu erzeugen. Er liefert Signale in drei Lautstärkestufen mit einer programmierbaren Ton- und Melodiefolge; sein Speicher faßt 512 Töne. Der NF-Ausgang wird auch für das ASCII-Glockensignal (BELL) und für den Tastenklick benutzt. Der Frequenzbereich umfaßt 6 Oktaven, aber man darf hier natürlich keine HiFi-Qualität erwarten.

I.2: Die Farbzusatzkarte GRIP-COLOR

GRIP-CCOLOR bietet für anspruchsvolle Grafikanwendungen eine Farb- bzw. Graustufenauflösung von 4 Bit pro Bildpunkt. Der Rot-, Grün- und Blauanteil eines Bildpunkts läßt sich dabei unabhängig in 16 Stufen einstellen, so daß sich insgesamt 16x16x16=4096 Farbtöne zusammenmixen lassen. Damit ist das ganze Spektrum abgedeckt; die Unterschiede zwischen den einzelnen Farbtönen sind kaum erkennbar, die Farben gehen praktisch kontinuierlich ineinander über.

Jeder Bildpunkt erhält seine Farbinformation aus dem auf das vierfache (256 KByte) erweiterten Bildschirmspeicher; sie wird über eine **Look-Up-Tafel** in eine wirkliche Farbe umgesetzt und sichtbar gemacht. Diese Methode hat natürlich auch einen Haken: Es lassen sich zwar 4096 Farben auf dem Bildschirm darstellen, davon aber immer "nur" 16 gleichzeitig, da ja nur 4 Bit pro Bildpunkt zur Verfügung stehen. Man kann allerdings mit einer

zweiten GRIP-COLOR-Karte die Zahl der gleichzeitig sichtbaren Farben auf 128 erhöhen.

Die Look-Up-Methode bietet aber zugleich auch einen bestechenden Vorteil: Die Farbe eines Bildpunkts wird nicht durch seine Farbinformation (**Farbindex**), sondern durch den zugehörigen Listeneintrag bestimmt, der sich durch eine einzige Schreiboperation ändern läßt. Man kann also alle Farben auf dem Bildschirm blitzschnell umdefinieren und damit das Aussehen des Bildes völlig ändern, ohne seinen Inhalt zu beeinflussen.
Insbesondere lassen sich unterschiedliche Farbindizes zunächst mit der gleichen Farbe belegen und dann plötzlich ändern; so kann man den Eindruck schneller Bewegung schaffen. Die Blink-Funktionen der GRIP-COLOR basieren auf diesem Prinzip.

Da auf alle Ebenen des Farb-RAM's gleichzeitig zugegriffen werden kann, ist die Farbgrafik keineswegs mit einem Geschwindigkeitsverlust verbunden; im Gegenteil geht durch die Zusatz-Hardware auf der COLOR-Karte das Zeichnen von farbigen Vektoren sogar schneller.

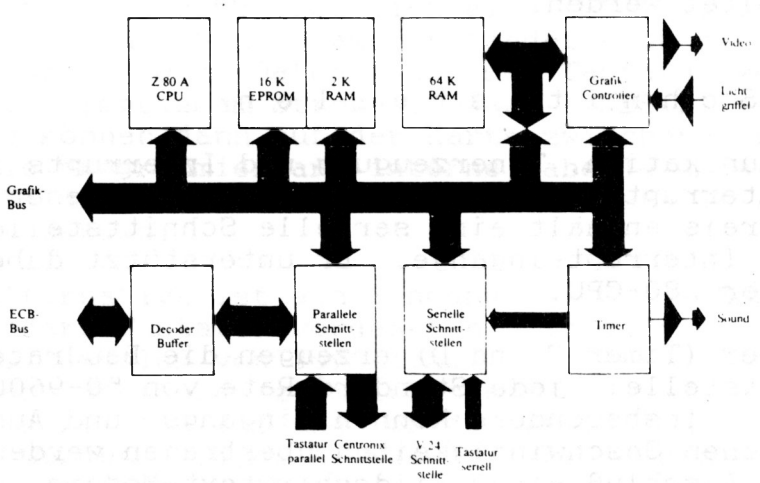
GRIP-COLOR wird entweder über den ECB-Bus (96-poliger Bus erforderlich!) mit der GRIP-Grundkarte verbunden oder als 'Huckepack'-Platine aufgesteckt.

II. --- Schaltung ---

Der Übersicht halber wurde der Schaltplan im Anhang in drei Funktionsgruppen aufgeteilt:

a) CPU-Teil mit EPROM, RAM und Schnittstellen,
b) Grafik-Teil mit Video-Controller und Bildspeicher für die Grundebene,
c) COLOR-Teil mit dem Farb-RAM und den Farbtafelregistern.

II.1: Schaltung der GRIP-Grundkarte



Ein aus drei Invertern (Z18-A,B,C) gebildeter Quarzoszillator erzeugt den zentralen 15-MHz-Systemtakt, von dem die meisten Signale auf der Karte abgeleitet sind. Dabei bilden die beiden über 270-Ohm-Widerstände "kurzgeschlossenen" Inverter einen Breitbandverstärker, der vom Quarz durch positive Rückkopplung zum Schwingen angeregt wird.

Die **Z80-CPU** (Z1) läuft mit 3.75 MHz Taktfrequenz, die über den Zähler **Z34** vom Systemtakt heruntergeteilt werden. Beim Einschalten erzeugt ein RC-Glied (C21) zusammen mit dem Schmitt-Trigger

Z11-A einen kurzen Reset-Impuls, der die CPU und die Portbausteine initialisiert. Als zeitbestimmender Widerstand für das RC-Glied wird der Eingangswiderstand des Schmitt-Triggers ausgenutzt.

Speicher und I/O-Ports werden von den Dekoder-IC's Z12 und Z13-A ausgewählt. Die Ports belegen dabei alle internen Adressen von 00h-7Fh; der Rest ist für die COLOR-Zusatzkarten reserviert. Das Betriebssystem und die verschiedenen Zeichensätze stehen in einem 32-KByte-EPROM vom Typ 27256 (Z2); ein statisches 8-KByte-RAM (Z3) bietet Platz für Stack, Parameter und einen Teil der Zeichenpuffer. Dieses Zusatz-RAM ist notwendig, damit die User-Zeichensätze batteriegepuffert werden können. Die UBAT-Leitung des ECB-Bus (a24) wird dazu benutzt.

Der Demultiplexer mit Latchausgängen, Z8, steuert acht Leitungen für die Kontrollflags /I, /DPE, /STR, PAGE, VOL0-1 und /RTS. Diese Flags beeinflussen Funktionen des Grafikcontrollers und der Schnittstellen (s.u.). Jedes Flag kann von der CPU über einen von acht Ports umgeschaltet werden.

S c h n i t t s t e l l e n

Für serielle Kommunikation, Tonerzeugung und Interrupts ist der STI(Serial Timer Interrupt)-Baustein Z9 zuständig. Dieser hochintegrierte Schaltkreis enthält eine serielle Schnittstelle, vier Zeitgeber und acht Interrupteingänge. Er unterstützt dabei die Vektor-Interrupts der Z80-CPU.

Zwei der Zeitgeber (Timer C und D) erzeugen die Baudraten für die serielle Schnittstelle; jede Standard-Rate von 50-9600 Baud ist programmierbar. Insbesondere können Eingangs- und Ausgangsdaten mit verschiedenen Geschwindigkeiten übertragen werden, was für den eventuellen Anschluß eines Bildschirmtext-Modems wichtig ist.

Für Hochgeschwindigkeits-Übertragungen gibt es noch eine "unübliche" Baudrate von 30000 Baud. Die Abweichungen der intern erzeugten Baudraten liegen unter 2%. Ein Taktsignal, das der 16fachen Baudrate entspricht, liegt am Ausgang BAUD an. Stattdessen kann aber auch ein externer Baudratentakt über diese Leitung eingespeist werden.

Die Treiber- und Empfängerbausteine der seriellen Schnittstelle (Z10,Z11) passen die Pegel an den V24/RS232-Standard an. Außer den Datenein- und -ausgängen RX und TX verfügt die Schnittstelle über Steuerleitungen für Quittungsbetrieb (RTS und CTS). Wenn diese Steuersignale nicht benutzt werden sollen, ist der CTS-Eingang mit +12V zu verbinden.

Z6, ein programmierbares Parallelinterface (PPI) von Typ 8255, erfüllt eine doppelte Aufgabe. Port A ist mit dem ECB-Bus verbunden und auf bidirektionalen Datentransfer eingestellt. Seine Handshake-Leitungen sind dabei durch eine Dekodierlogik (**Z13-B, Z15-A, Z16, Z17-B**) mit den Bussignalen verknüpft, so daß der Port wie eine normale I/O-Einheit auf zwei Adressen vom ECB-Bus aus ansprechbar ist.

Daten, die von "außen", d.h. vom Bus, in den Port A eingeschrieben werden, lassen sich "innen" aus dem 8255 wieder auslesen. Das Ganze funktioniert natürlich auch in umgekehrter Richtung. Sobald ein eingeschriebenes Datenbyte von der jeweiligen Empfänger-CPU gelesen wurde, erfährt die Sender-CPU dies über ein Statussignal und kann jetzt das nächste Byte übermitteln. Zur Beschleunigung des Transfers kann über die STI ein Interrupt ausgelöst werden, sobald eines dieser Bits auf "1" springt.

An Port B des 8255 läßt sich eine ASCII-Tastatur mit Parallelausgang anschließen. Dazu befindet sich der Port im "strobed input"-Modus. Die Daten von der Tastatur werden durch einen negativen Impuls am Strobe-Eingang (**/KBSTB**) von Port B übernommen. Sie können dann auf der Karte zwischengespeichert, eventuell durch die programmierbare interne Tabelle umgesetzt und dann über den ECB-Bus oder die serielle Schnittstelle an den Hostrechner gesendet werden.

Als Alternative ist ein Eingang des STI (**SKBD**) für eine serielle Standard-Tastatur vorgesehen; in diesem Fall wird Port B des 8255 zum Anschluß von Joysticks o.ä. frei. Da die Baudrate von seriellen Tastaturen ziemlich niedrig ist (i.a. 600 Bd), kann die Seriell-parallel-Wandlung von der CPU, d.h. per Software, übernommen werden. Timer A erzeugt hierfür den Baudratentakt durch periodische Interrupts.

Für eine IBM-kompatible serielle Tastatur wird der **/NMI**-Eingang der CPU als Takteingang, der **I6**-Eingang der STI als Dateneingang benutzt.

Der Tongenerator benutzt den STI-Timer B, zwei Kontrollflags und die freien Ausgangstreiber (Z10-C,D). Die Schaltung erzeugt Töne beliebiger Frequenz in drei Lautstärkestufen; die Lautstärke läßt sich dabei über die oben erwähnten Flags **VOL0-1** einstellen.

Das Centronics-Interface dient der Ansteuerung eines Druckers mit der üblichen parallelen Schnittstelle. Es wird von dem 8-Bit-Parallelport **Z7** und einigen Steuerleitungen (**/STB, BUSY, /INIT, /ERROR**) gebildet. **/STROBE** fordert mit einem "0"-Impuls zur Übernahme des nach Z7 eingeschriebenen Datenbytes auf.

Die BUSY-Rückmeldung ist auf den STI geführt und löst einen Interrupt aus, sobald der Drucker zum Empfang des nächsten Bytes bereit ist. /ERROR meldet einen Fehler des Druckers. Für spezielle Anwendungen können die 8 Datenleitungen der Schnittstelle (DATA1-8) über den Jumper J2 in den hochohmigen Zustand versetzt werden.

V i d e o - L o g i k

Die zweite Funktionsgruppe der Karte enthält den 64-KByte-Hauptspeicher, der von CPU und Grafik-Controller (Z30) gemeinsam benutzt wird. Als Controller wird ein Video-Steuerbaustein (6845) verwendet, der eigentlich für textorientierte Bildschirme entworfen wurde. Er läßt sich jedoch auch gut für Grafik-Zwecke einsetzen.

Die für Textverarbeitung wichtige Übertragung von Zeichen oder Symbolen erfolgt im übrigen mit dem 6845 schneller als mit speziellen Vektorprozessoren, weil die CPU hier direkt auf den Bildschirmspeicher zugreifen kann. Außerdem ist der Speicher speziell für schnelle Zeichenübertragung per LDIR-Befehl organisiert (Blockstruktur, s.u.). Daher ist GRIP bei der Textübertragung mindestens genauso schnell wie ein reines, nicht grafikfähiges Text-Videointerface, das mit einem Zeichengenerator arbeitet.

Für die CPU sieht der 6845 wie ein I/O-Baustein mit 18 internen Registern aus. Mit Hilfe der Register kann unter anderem das Bildschirmformat, die Zeilenfrequenz und -Anzahl und der Adressbereich des Grafik-Speichers programmiert werden. Durch entsprechende Einstellung lassen sich Teile des Bildes dunkelschalten oder verschieben (Scrolling).

B i l d s p e i c h e r

Der Hauptspeicher ist in zwei Seiten zu je 32 KByte eingeteilt, die den oberen Adressbereich der CPU (8000h-FFFFh) belegen. Die Seiten werden durch das PAGE-Flag umgeschaltet.

Normalerweise besteht der Speicher aus acht dynamischen 64-KB-RAM's. Durch Einsatz von 256-KB-RAM's kann der Bildspeicher auf der Grundkarte auf 128 KByte ausgebaut werden; die I/O-Leitung I2 der STI bedient dabei über den Jumper J13 die zusätzliche Adressleitung. Dieser Ausbau ist aber nur sinnvoll, um z.b. einen größeren Spooler oder weitere Bildschirmseiten zu erhalten; noch höhere Grafikauflösung kann damit nicht erreicht werden. Zur Zeit (März 1986) wird ein 128-KB-Bildspeicher von der Software noch nicht unterstützt.

Die Multiplexer Z4,Z5,Z28,Z29 schalten die RAM-Adressen und die Zugriffssignale synchron zwischen CPU und Grafik-Controller um. Für das richtige Timing und das notwendige ständige Auffrischen (Refresh) der Speicherbausteine sorgt das Flipflop **Z35-A**. Das Schieberegister Z36 liest ständig die gerade angesprochenen Bildschirmadressen aus, wandelt die Daten in die serielle Form um und gibt sie über zwei Inverter an den Video-Ausgang aus. Das Video-Signal steht mit TTL-Pegel an dem Ausgang GRN zur Verfügung, mit 1-Volt-Pegel am Ausgang BAS. Durch Verändern der Widerstände R7-R9 lassen sich die Pegel abstimmen.

Die Bildpunkte (Pixels) sind in etwas eigenwilliger Weise den Bits im Speicher zugeordnet. Der Bildschirm besteht aus rechteckigen Blocks von 8x8 Pixels; eine zuschaltbare neunte oder zehnte Blockzeile (-> ESC ESC '6') bleibt immer dunkel. Acht aufeinanderfolgende Bytes bilden von oben nach unten die Pixelzeilen des Blocks, beginnend mit Bit 0 des ersten Bytes. Diese Blockstruktur hat den Vorteil, daß ein Text- oder Grafikzeichen mit 8x8-Matrix durch einen einzigen, schnellen LDIR-Befehl in den Bildschirmspeicher eingeschrieben werden kann.

Das ganze Bild besteht aus 3360 (96x35) solcher zeilenweise aufeinanderfolgenden Blocks; dies entspricht einer Auflösung von 215040 (768x280) Bildpunkten. Im Zeilensprung-Verfahren (interlaced) verdoppelt sich die Anzahl der Blocks auf 6720.

Das Zeilensprung-Verfahren wird z.b. beim Fernsehen benutzt, hat aber wegen der dabei auf 25 Hz reduzierten effektiven Bildwechselfrequenz einen entscheidenden technischen Nachteil: Das Bild flimmert. Beim Fernsehen mit seinen bewegten Bildern fällt das nicht so auf, aber für Computerterminals ist es ergonomisch ungünstig (Kopfschmerzen!). Daher ist nur ein Bildschirm mit langer Nachleuchtdauer (mit **P39-Phosphorbeschichtung**) für dieses Verfahren geeignet, z.b. **CRT-200** von CONITEC.
Auch ohne Zeilensprung erzielt man mit nachleuchtenden Monitoren bessere Ergebnisse.

L i c h t g r i f f e l

Eine Besonderheit des 6845 ist der Lichtgriffel-Eingang (LP). Bei einer positiven Flanke an diesem Eingang wird die augenblickliche Bildschirmadresse in zwei internen Registern zwischengespeichert. Ein Lichtgriffel besteht im allgemeinen aus einem Stift mit einem Phototransistor in der Spitze.

Wird der Stift auf eine helle Stelle des Bildschirms gesetzt, so verringert sich der Widerstand des Sensors, wenn der Elektronenstrahl diese Stelle überstreicht. Der Transistor ist zwischen

den /LPEN-Eingang und Masse zu schalten; R15 setzt die Widerstandsänderung in einen LOW-Impuls um.

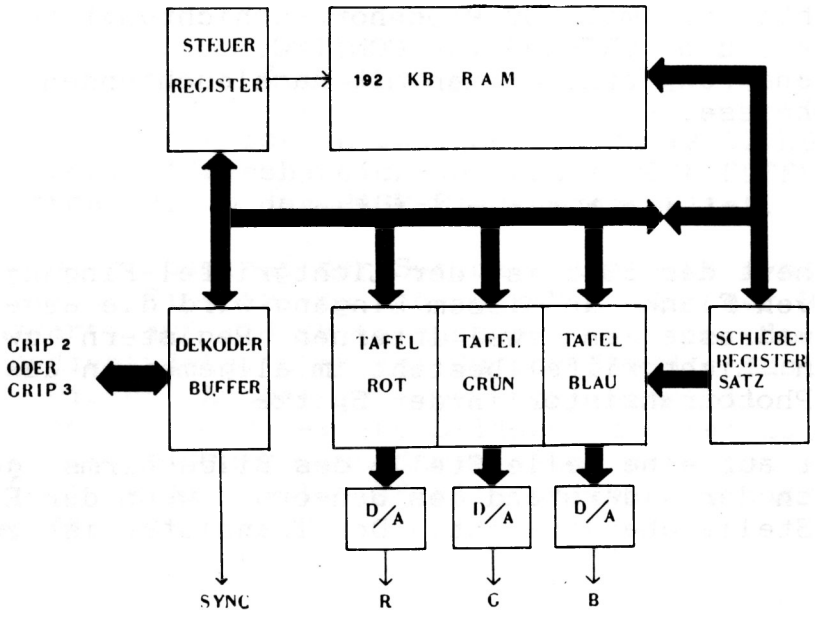
Dieser Lichtgriffel-Impuls wird durch den Timer-Komparator Z31 aufbereitet und digitalisiert. Die Ansprechschwelle läßt sich dabei mit dem Trimmwiderstand TR1 einstellen. Liegt der Impuls über der Schwelle, löst er über den LP-Eingang des 6845 und über das Tristate-Latch Z33 die Speicherung der gerade angesprochenen Bildschirmadresse aus.

Die Auflösung beträgt vier Bildpunkte in der horizontalen und zwei in der vertikalen Richtung. Nach dem Auslösen ist der Lichtgriffel-Detektor für weitere Impulse solange gesperrt, bis er durch einen I/O-Befehl über das Signal LRS wieder "scharf" gemacht wird.

Der 6845 besitzt zusätzlich einen Cursor-Ausgang, der einen Impuls erzeugt, sobald der Elektronenstrahl der Bildröhre eine durch ein internes Register bestimmte Stelle auf dem Bildschirm überstreicht. Dieses Signal wird über den Jumper J13 auf einen Eingang des STI geführt, so daß bei Erreichen einer beliebigen Bildschirmposition ein Interrupt ausgelöst werden kann. Dieses Feature wird jedoch in der jetzigen Betriebssoftware nicht benutzt.

II.2: Schaltung der GRIP-COLOR

Das Blockschaltbild zeigt das Prinzip der Farberzeugung:



Der Zugriff auf die Ports der Karte erfolgt von der CPU auf der GRIP-Grundkarte, und nicht etwa vom ECB-Bus. Dadurch läßt sich der Farbzusatz auch benutzen, wenn GRIP über die serielle Schnittstelle mit dem Hostrechner verbunden ist. Der Dekoder/ Bufferblock selektiert die einzelnen Ports auf der COLOR-Karte und puffert die Daten- und Adreßssignale.

Das Steuerregister kontrolliert den Zugriff auf die 192 KByte des Farbspeichers, in den parallel zur Grundebene eingeschrieben werden kann. Der Inhalt des Speichers wird zur Auffrischung des Farbbildes ständig auf den Schieberegisterblock ausgegeben und in drei serielle Datenströme umgewandelt. Außerdem kann der Speicher über einen Satz Tristate-Puffer von der Grundkarte ausgelesen werden.

Die Video-Datenströme gelangen nun in die programmierbare "Colour-Look-Up-Tafel", die für die Farberzeugung zuständig ist. Sie erzeugt ein vier-Bit-Signal für jede der drei Grundfarben, das von den drei nachgeschalteten D/A-Wandlern in ein hochfrequentes Analogsignal umgewandelt wird. Die drei Analogsignale für Rot, Grün und Blau bilden das endgültige Farbsignal und können direkt zur Ansteuerung eines Farbmonitors verwendet werden.

Um die Funktionen im einzelnen zu verstehen, betrachten wir nun den Schaltplan. GRIP-COLOR enthält 49 I/O-Register, die auf 4 Portadressen angesprochen werden können. Eins davon - das zentrale 8-Bit-Steuerregister - steuert die Übertragung der Bildinformation zum bzw. vom Video-RAM. Mit den restlichen 48 4-Bit-Farbtafelregistern werden die Farbtöne über die Intensität des Rot-, Grün- und Blauanteils eingestellt.

Daß mit nur drei Adressen 48 Register bedient werden können, klingt zunächst paradox. Es handelt sich jedoch um lediglich um 4-Bit-Register, so daß von der eingeschriebenen 8-Bit-Information vier Bit zum Adressieren verwendet werden können. So lassen sich über jede der drei Portadressen 2 hoch 4 = 16 Register ansprechen.

Zum Ansteuern dienen die Signale des Grafikbusses. Der Dekoder Z7 wählt die Ports aus; dazu werden die "gemultiplexten" Adressen XA0-7 benutzt, die auch das RAM adressieren. Während des I/O-Zugriffs der Z80 liegen die unteren 7 Adressbits auf dem Multiplex-Adressbus und können vom Dekoder ausgewertet werden.

Die 4 Ports liegen normalerweise auf den internen Adressen C0h, D0h, E0h und F0h. Wenn der Jumper J2 in Position B gesteckt wird, ändern sich die Adressen auf 80h, 90h, A0h und B0h; dies ist bei

Verwendung von zwei kaskadierten COLOR-Karten erforderlich.

Die Farbinformation für das Bild wird in die 24 dynamischen RAM's Z15-23, Z26-34 und Z36-41 eingeschrieben. Die Karte enthält 24x64 KBit = 192 KByte RAM; zusammen mit der Grundplatine GRIP stehen also 256 KByte RAM in vier 64-KB-Ebenen als Videospeicher zur Verfügung. Für jeden der 768x280 Bildpunkte (bzw. 768x560 im Zeilensprungverfahren) gibt es damit 4 Bit Farbinformation, so daß sich 16 verschiedene Farben gleichzeitig auf dem Schirm darstellen lassen. Mit einer zweiten COLOR-Karte erhöht sich die Zahl der Bits pro Bildpunkt auf 7 und damit die Zahl der gleichzeitig sichtbaren Farben auf 128.

Die Organisation des Farb-Bildspeichers ist identisch mit dem Grund-RAM auf GRIP. Gleiche Adressen entsprechen gleichen Bildschirmpunkten. Auch die Umschaltung über das PAGE-Flag funktioniert auf der Zusatzkarte genauso wie auf der Grundkarte.

Einen wesentlichen Unterschied gibt es beim Einschreiben in den Farbspeicher. Während auf der Grundkarte jedes Byte direkt in den Speicher geschrieben wird, hat es beim Farb-RAM lediglich die Funktion einer Maske. Nur die Bits, die auf "1" gesetzt sind, geben über die NAND-Gatter Z14 und Z35 das Schreiben in die entsprechenden Bitpositionen frei. Die Bitpositionen, die einer "0" im Maskenbyte entsprechen, bleiben ungeändert.

Es wird immer in alle drei Farb-Ebenen gleichzeitig geschrieben. Der neue Wert für veränderten Bits in jeder Ebene wird mit drei Flags (COL0, COL1, COL2) im Kontrollregister auf F0h (Z25) eingestellt. Dieser Wert entspricht drei Bits des 4-Bit-Farbindex im RAM; das vierte Bit kommt aus der Grundebene.

Diese **Maskenmethode** hat den großen Vorteil, daß für jede Farbe nur ein einziger Schreibzyklus erforderlich ist; die zuletzt eingeschriebene Farbe überdeckt dabei immer die vorherige. Damit ist die Farbgrafik theoretisch genauso schnell wie die Schwarzweißgrafik mit der Grundkarte. In der Praxis ist sie sogar noch schneller! Beim Vektorschreiben kann nämlich das Einlesen und "Verodern" des vorherigen Speicherinhalts entfallen, wodurch jeder Vektor im Farbmodus um 15% schneller gezeichnet wird.

In die Grund- und in die Farbebenen kann gleichzeitig geschrieben werden. Der Schreibmodus wird über die Flags WE0-1 im Steuerregister eingestellt. WE0 steuert das Schreiben in die Farb-, WE1 in die Grundebene; eine "1" gibt das Schreiben frei.

Das Lesen aus dem Farb-RAM ist ebenfalls möglich und erfolgt getrennt für jede Ebene, genauso wie auf der Grundkarte. Aus welcher der vier Ebenen gelesen wird, bestimmen die Flags RE0-2 im Steuerregister.

Zum Anschluß eines RGB-Farbmonitors (oder eines monochromen Monitors für Graustufen) steht der Stecker N2 zur Verfügung. Der Schwarzweißmonitor an der Grundkarte kann nach wie vor angeschlossen bleiben. Der gleichzeitige und völlig unabhängige Betrieb von zwei Monitoren ist problemlos möglich und in Spezialfällen sogar sinnvoll: Ein Monitor für Text, der zweite für Grafik.

III. -- Aufbau und Inbetriebnahme

III.1: Hardware-Aufbau

Die Selbstbestückung der Karte ist wegen der IC-Dichte (fast tausend Lötstellen) auch für Spezialisten nicht ganz einfach. Wenn man noch nicht viel Erfahrung hat, ist vom Selbstbau grundsätzlich abzuraten. Bei sorgfältigem Vorgehen spricht allerdings nichts dagegen, daß das Gerät beim Einschalten Anhieb funktioniert.

Die Inbetriebnahme sollte trotzdem stufenweise erfolgen, damit eventuelle Fehler sofort lokalisiert werden können. Ein hochohmiger Lautsprecher, ein Labornetzteil, ein Multimeter und vor allem ein gutes Zweistrahl-Oszilloskop (mindestens 20 MHz) sind zur Fehlersuche unbedingt erforderlich.

ACHTUNG: Bevor Sie den Lötkolben in die Hand nehmen, lesen Sie dieses Handbuch gründlich durch! Machen Sie vorher nicht weiter!

A. Am Anfang steht das Löten. Man sollte am besten mit einem Schwallbad, wenigstens aber mit einer temperaturgeregelten Lötstation arbeiten und alle 30 Minuten eine kurze Pause einlegen, um sich zu entspannen und das bisherige Werk zu kontrollieren.

Zuerst sind alle IC-Sockel und die Steckerleisten einzulöten; anschließend kommen die Widerstände, der Trimmer, die Kondensatoren und der Quarz an die Reihe. Pin 1 aller IC's zeigt zur VG-Leiste oder zum Tastaturstecker. Die IC's werden noch nicht in die Sockel eingesetzt!

Beim Einbau des ECB-Bus-Steckers (VG-Leiste) muß man sich jetzt zwischen einem 64- oder 96-poligen Typ entscheiden, je nach Bus-Version und späterem Anschluß einer COLOR-Karte (s.u.).

Fast alle später eventuell auftretenden Fehler werden in dieser Phase gemacht. Achten Sie bitte insbesondere darauf, daß kein Lötzinn durch die Bohrungen auf die Oberseite der Platine gelangt. Wenn die Lötarbeit beendet ist, wird es erst richtig spannend: Jetzt werden die Funktionen der Karte der Reihe nach ausgetestet.

B. Alle Sockel sind noch leer. Nun wird die 5-Volt-Betriebsspannung über den ECB-Bus-Stecker angelegt und die Stromaufnahme gemessen. Sie muß - nach einem kurzen Sprung zum Aufladen der Kondensatoren - etwa 10 mA betragen (über das Trimmpoti), sonst ist etwas faul. Kurzschlüsse und verkehrt gepolte Elektrolytkondensatoren machen sich jetzt i.a. mit Rauchsignalen bemerkbar.

Wenn alles soweit o.k. ist, sind sämtliche Versorgungspin's der IC-Sockel anhand des Schaltplans auf korrekte Betriebsspannung zu kontrollieren. Bei den dynamischen RAM's liegen Masse und +5 Volt gegenüber den TTL-IC's genau andersherum!

C. Als erstes IC wird der 74S04 (Z18) in den Sockel gesteckt. Vor jedem IC-Einsetzen die Spannung ausschalten! Nach dem Wiedereinschalten muß an den Pin's 4 und 6 des 74S04 ein symmetrisches 15-MHz-Signal anliegen.

D. Die IC's Z11 (1489) und Z34 (74HC163) werden nun bestückt. An Pin 6 des CPU-Sockels (Z1) sollte ein kräftiges 4-MHz-Taktsignal erkennbar sein, Pin 26 muß auf HIGH liegen.

E. Jetzt kann der CPU-Teil in Betrieb genommen werden. Zu bestücken sind vorerst nur die CPU (Z1), das EPROM (Z2) und 3 IC's zur Dekodierung, nämlich Z12, Z13 und Z15 (74HC138, 74HC139, 74HC32). ACHTUNG - Z2 und Z3 sind gegenüber den anderen IC's um 180 Grad gedreht! Vorher sind noch die Default-Positionen der Jumper zu beachten:

J1: Vorverbunden auf 2-3 (27128/256-EPROM)
J2: Vorverbunden auf 1-2,3-4 (Centronics eingeschaltet)
J3a,b: siehe -> III.4, -> III.5
J4: gesteckt! (keine COLOR-Zusatzkarte)
J5: Vorverbunden (Reset intern)
J6: Vorverbunden auf 1-2,3-4 (Baudraten intern)
J7: Vorverbunden auf 1-2,3-4 (ECB-Bus-Adressen C0h-C1h)
J10: Vorverbunden (Takt intern)
J11: gesteckt auf 2-3 (27256-EPROM)
J12: gesteckt auf 1-2 (6264-RAM)
J13: offen (Position egal)

Beim Einschalten muß jetzt bereits ein Programm ausgeführt werden: Solange der Video-Steuerbaustein nicht auf der Karte ist, springt das Betriebssystem in eine Testroutine, bei der unter anderem zyklisch alle Ports angesprochen werden. Folglich sind auf den SELECT-Eingängen der I/O-Chip's periodische LOW-Impulse von etwa 700 ns Dauer zu erwarten.

Diese Impulse sind abgreifbar an folgenden IC-Pin's: Z6-6, Z7-11, Z8-14, Z9-30, Z14-1, Z33-1,2, Z30-23. Bei dem letzten IC, dem 6845, sind die Select-Impulse invertiert.

Wenn nichts zu sehen ist, sollte man zuerst versuchen, durch mehrfaches Rücksetzen der CPU (mit einem Stück Draht zwischen Pin 26 und 29) das Programm doch noch zu starten. Gelingt es,

liegt der Fehler an der RESET-Leitung oder an dem Kondensator C21; andernfalls sind alle Signale am EPROM zu überprüfen. Kurzschlüsse und Unterbrechungen der Daten- und Adressleitungen lassen sich im allgemeinen bereits an der Signalform erkennen.

Die Pegel auf den Adressleitungen müssen sauber und ausgeprägt sein. Eine Datenleitung, deren Signalform von denen der 7 anderen wesentlich abweicht, ist ebenfalls sofort verdächtig. Ohne Oszilloskop führt kein Weg daran vorbei, bei einem Fehler alle IC's wieder zu entfernen und die Leitungen mit dem Ohmmeter "durchzuklingeln" bzw. auf Kurzschlüsse, z.b durch unsauberes Löten, zu kontrollieren.

F. Jetzt kann ein kleiner Lautsprecher oder Kopfhörer an den SOUND-Ausgang angeschlossen werden. Nach Bestücken der IC's Z8 (74HC259), Z10 (1488) und der STI (Z9) muß nach dem Wiedereinschalten ein 1000-Hz-Pfeifton zu hören sein.

G. Das Schwierigste ist geschafft. Nun werden alle restlichen IC's bis auf Z32 (74LS244) und den VDC-Baustein Z30 eins nach dem anderen in die Fassungen gesetzt. Nach jedem Einsetzen sollte kontrolliert werden, ob der Pfeifton noch zu hören ist.

H. Nach Einsetzen von Z32 muß der Pfeifton um eine Oktave abnehmen. Daran läßt sich das Funktionieren des 64-KByte-Hauptspeichers erkennen.

I. In dieser letzten Phase kann nach Bestücken von Z30 endlich der Video-Monitor, wie unten beschrieben, angeschlossen werden. Nach Einschalten der Karte erscheint der blinkende Cursor auf dem Bildschirm. Oben sieht man die Statuszeile; die Uhr ist nicht sichtbar, solange die Zeit noch nicht eingestellt wurde.

Die Karte läßt sich nach erfolgreichem Aufbau über die Tastatur im LOCAL-Modus (s.u.) ausprobieren. Jumperstellung (J3) beachten!

III.2: Anschluß des Monitors

An die Grundkarte (Stecker N2) läßt sich ein monochromer Monitor anschließen, dessen Bandbreite wegen der hohen Auflösung möglichst über 18 MHz liegen sollte. Es empfiehlt sich generell, einen Monitor mit nachleuchtendem P39-Phosphor zu verwenden, z.b. CRT-200 von CONITEC. Im Interlaced- oder Pseudo-Interlaced-Betrieb ist dies unbedingt erforderlich, da sonst das Bild flimmert.

Es sind positive (HSYN,VSYN) und negative Synchronisationssignale (/HSYN,/VSYN) vorhanden. Normalerweise werden die Video- und Synchronisationssignale getrennt zugeführt. /HSYN und /VSYN lassen sich jedoch auch durch einfaches Verbinden zu einem /CSYN-Signal kombinieren (Composite Sync).

Zur Ansteuerung eines Monitors mit kombiniertem Video- und Synchronisationssignal (**BAS-Eingang**) müssen die drei Ausgänge BAS, /HSYN und /VSYN zur Erzeugung des BAS-Signals miteinander verbunden werden. Sie sind mit Open-Kollektor-Treibern versehen. Die Pegel können gegebenenfalls durch Ändern der Widerstände R7 und R9 angepaßt werden.

III.3: Anschluß von Lautsprecher/Verstärker

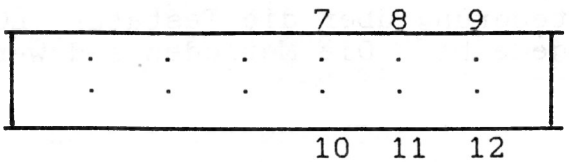
An den SOUND-Ausgang (Stecker N2, N3 oder N4) kann ein hochohmiger Lautsprecher (180 Ohm) oder ein NF-Verstärker angeschlossen werden. Das Signal kommt mit 24-Volt-Amplitude, bei Verwendung eines Verstärkers sollte deshalb ein Spannungsteiler dazwischengesetzt werden.

Die Eingangsimpedanz sollte, eventuell durch Parallelschalten eines Widerstands, so abgestimmt werden, daß sich die drei Lautstärkestufen deutlich unterscheiden lassen. Notfalls ist **R17** zu ändern.

III.4: Anschluß der Tastatur(en)

Es lassen sich drei verschiedene Tastaturtypen an GRIP anschließen; damit dürfte es wohl kaum eine Tastatur am Markt geben, die sich nicht anschließen läßt. Während des Betriebs kann die Tastatur durch einen Host-Befehl (-> **ESC ESC '4'**) gewechselt werden. Beim Einschalten bestimmt Jumper **J3b**, welche Tastatur aktiv ist:

J3b offen: Paralleltastatur -> III.4.1
7-8: Serielle Tastatur, 1200 Baud, 8 Bit -> III.4.2
8-9: Serielle Tastatur, 1200 Baud, 7 Bit -> III.4.2
11-12: Serielle Tastatur, 600 Baud, 7 Bit -> III.4.2
10-11: IBM-Tastatur, asynchron, 8 Bit -> III.4.3



III.4.1. **Paralleltastatur**: Hierzu dient der Stecker N4. Normalerweise werden nur die 7 Datenleitungen **KBD0-KBD6** und die negative STROBE-Leitung **/KBDSTB** (und natürlich +5V und GND) benötigt. Die achte Datenleitung **KBD7** ist, wenn sie nicht angeschlossen wird, auf Masse zu legen.

Sendet die Tastatur invertierte Daten, so muß entweder ein Inverter (z.b. 74LS240) zwischengeschaltet oder GRIP durch einen Befehl vom Host (-> **ESC ESC '4'**) darauf angepaßt werden.

Der Ausgang **KBF** (Keyboard Buffer full) ist als Handshake-Leitung für eine Tastatur mit internem Puffer gedacht und bleibt normalerweise offen (ist auch unnötig, da GRIP selbst bereits einen Tastaturpuffer besitzt). Er kann jedoch als BUSY-Leitung verwendet werden, wenn - für irgendeine exotische Spooler-Anwendung - der Paralleltastatureingang an einen Centronics-Ausgang angeschlossen wird.

III.4.2. **Serielle Tastatur**: Stecker **N3**, Leitung **SKBD**. Es können sowohl V24- wie auch TTL-Signale angelegt werden; GRIP erkennt beim Einschalten automatisch die Polarität. Die Anfangs-Baudrate wird über **J3b** eingestellt (s.o.). Andere Baudraten lassen sich vom Host aus (-> **ESC ESC '3'**) einstellen.

III.4.3. **IBM-ähnliche Tastatur**: Datenleitung an **KDATA**, Taktleitung an **KCLOCK**, Stecker **N4**. Eventuell, je nach Tastatur, muß an KDATA noch ein Pull-up-Widerstand (2.2 kOhm nach +5V) angeschlossen werden.

ACHTUNG: die Tastatur sollte ASCII-Codes aussenden! 'Echte' IBM-Tastaturen lassen sich zwar anschließen, jedoch ist in die GRIP-Software z.Zt. noch keine IBM -> ASCII-Umcodierung eingebaut.

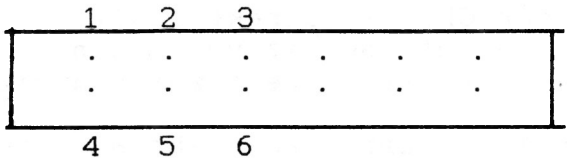
III.5: Anschluß des Hostrechners

Es gibt vier verschiedene Möglichkeiten, GRIP an Ihren Rechner anzuschließen:

a) über den ECB-Bus -> III.5.1
b) direkt an den Datenbus der CPU -> III.5.2
c) über eine V24/RS232-Schnittstelle -> III.5.3
d) über einen der drei Tastaturports -> III.4

Die letzte Methode - Ansteuerung über die Tastatur (LOCAL-Modus) - ist nur zu Testzwecken gedacht. Die Methoden a-d werden über Jumper **J3a** ausgewählt:

J3a offen: Host = ECB- oder Datenbus -> III.5.1/2
1-2: Host = V24, 9600 Baud -> III.5.3
2-3: Host = V24, 4800 Baud -> III.5.3
4-5: Host = V24, 1200 Baud -> III.5.3
5-6: Host = Tastatur (LOCAL-Modus) -> III.4



III.5.1. **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 der GRIP-COLOR enthält (-> III.7). Wenn also ein 96-poliger Bus und -Stecker benutzt wird, darf die b-Leiste auf diesem Abschnitt nicht anderweitig belegt sein!

Das Bus-Timing ist unkritisch; GRIP kann mit Busfrequenzen bis **8 MHz** (Z80H) betrieben werden. Wenn der ECB-Bus niederohmig terminiert ist, kann eventuell die Treiberleistung des Portbausteins für das Abfragen des Datenports nicht mehr ausreichen. Für solche Fälle sollte ein Zusatz-Buffer vom Typ 74LS245 für den Datenbus eingesetzt werden. Ein bereits vorverdrahteter Platz dafür ist unter **Z6** vorgesehen. Der mittlere Steg des Sockels muß dazu vorsichtig entfernt, die acht mit kleinen Quadraten markierten Datenleitungen auf der Oberseite der Platine durchgetrennt werden.

Der **/PCL**-Ein/Ausgang dient zum Rücksetzen der GRIP. Beim Einschalten wird dort ein kurzer LOW-Impuls über einen Open-Kollektor-Treiber ausgegeben (interner Reset). Ist dies nicht erwünscht, muß Jumper **J5** aufgetrennt werden.

Die Host-CPU spricht GRIP über zwei I/O-Adressen an, eine für Daten, die zweite für den Status. Die Adressen lassen sich über Jumper J7 einstellen:

J7 1-3,2-4: Adressen **C0h-C1h** (default, vorverbunden)
1-2,3-4: Adressen **A0h-A1h** (Default-Verbindung auftrennen!)

Wegen der Vorverbindung durch dünne Leiterbahnen auf der Unterseite muß dieser Jumper nur berücksichtigt werden, wenn die Adressen C0h und C1h im System schon anderweitig belegt sind.

Auf der zweiten Adresse (Datenport, R/W, normalerweise C1h) werden die Daten mit I/O-Befehlen eingeschrieben oder ausgelesen, auf der ersten (Statusport, R/O, C0h) der Status abgefragt. Beim

Status sind nur die Bits 6 und 7 relevant.

	D7	D6	D5	D4	D3	D2	D1	D0
Datenport:								
Statusport:	RF	WE	x	x	x	x	x	x

D0-D7: Datenleitungen zur GRIP (bidirektional)
RF : Read Buffer Full - GRIP bereit zum Lesen
WE : Write Buffer Empty - GRIP bereit zum Schreiben

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

Zur Verdeutlichung des Ganzen ein Unterprogramm im Z80-Code zur Ausgabe des Zeichens in Register C auf den Bildschirm:

```
GRIPOUT:  IN   A,(0C0H)   ;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 Register A, geht ähnlich:

```
GRIPIN:   IN   A,(0C0H)   ;Status abfragen
          BIT  7,A        ;Stehen Daten bereit?
          JR   Z,GRIPIN   ;Wenn nicht, das Ganze noch einmal
          IN   A,(0C1H)   ;Byte holen
          RET
```

Zur Vervollständigung einer CP/M-BIOS-Implementation noch die entsprechenden Status-Routinen:

```
GRIPIST:  IN   A,(0C0H)   ;Status
          BIT  7,A        ;Daten von Tastatur bereit?
          LD   A,0FFH     ;Falls ja, Return mit A=FF
          RET  NZ
          XOR  A          ;Falls nein, Return mit A=00
          RET

GRIPOST:  IN   A,(0C0H)   ;Status
          BIT  6,A        ;Fertig zur Ausgabe auf Bildschirm?
          LD   A,0FFH     ;Falls ja, Return mit A=FF
          RET  NZ
          XOR  A          ;Falls nein, Return mit A=00
          RET
```

III.5.2. Anderer Datenbus: Auch hier erfolgt der Anschluß über **N1**; GRIP belegt zwei Adressen im I/O- oder Memory-Bereich der Host-CPU. Jumper **J7** bleibt ungeändert.

/PCL ist ein Open-Kollektor-Ein/Ausgang zum Rücksetzen der Karte, der beim Einschalten einen kurzen, über **J5** abschaltbaren LOW-Impuls abgibt (-> III.5.1). Der Datenbus wird an D0-D7 angeschlossen. Zum Selektieren der Karte sind folgende Signalpegel erforderlich:

/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). **/RD** und **/WR** sind LOW-aktive Lese- und Schreibimpulse von mindestens 300 ns Länge.

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

Die Software-Ansteuerung erfolgt wie unter III.5.1 beschrieben.

ACHTUNG: Beim Anschluß von 8086-CPU's (z.b. ct86) ist zu beachten, daß bei ungeraden I/O-Adressen die Daten auf der oberen Hälfte des Adressbus ausgegeben werden! Darum sollte die Adressleitung A0 dieser CPU nicht an A0 der GRIP angeschlossen, sondern z.b. mit A1 vertauscht werden (-> c't 5/85, Seite 10).

III.5.3. V24-Schnittstelle: Anschluß an Stecker **N3**. Die Baudraten werden über Jumper **J3b** eingestellt (s.o.) und können außerdem, ebenso wie das Datenformat, vom Host umprogrammiert werden (-> ESC ESC '0' usw.). Das Default-Datenformat beim Einschalten ist:

8 Datenbits, 1 Stopbit, keine Parität.

Alle Standard-Baudraten zwischen 50 und 9600 Baud sind für Sender und Empfänger getrennt einstellbar. Für Hochgeschwindigkeits-Übertragungen gibt es noch eine "unübliche" Baudrate von 30000 Baud. Die Abweichungen der intern erzeugten Baudraten liegen unter 2%.

Ein Taktsignal, das der 16fachen Baudrate des V24-Empfängers

entspricht, liegt am Ausgang **BAUD** an. Stattdessen kann aber auch ein externer Baudratentakt (16xBaudrate) über diese Leitung eingespeist werden. Das geht über Jumper **J6**:

J6 1-2,3-4: Baudraten intern, BAUD ist Ausgang (vorverbunden).
2-4: Sender-Baudrate intern, Empfänger-Baudrate extern.
1-2: Baudraten (Sender + Empfänger) extern über BAUD.

Außer den Datenein- und -ausgängen RX und TX verfügt die Schnittstelle über Steuerleitungen für Handshake-Betrieb (RTS und CTS). Wenn diese Steuersignale nicht benutzt werden sollen, ist der CTS-Eingang mit +12V zu verbinden!
Die V24-Schnittstelle besitzt zwar im Host-Betrieb einen internen Datenpuffer von 1 KByte Länge, trotzdem sollte für Baudraten ab 30.000 Baud oder für zeitintensive Operationen (Flächenfüllen, Vektorzeichnen) RTS/CTS-Handshake oder XON/XOFF-Protokoll (-> ESC ESC '3') benutzt werden.

III.6: Anschluß des Druckers

III.6.1. Centronics-Schnittstelle: Anschluß an Stecker N3. Es sind nur die Signale **DATA1-DATA8, /STB, BUSY** und **GND** erforderlich. **/INIT** ist mit der /PCL-Leitung des ECB-Bus verbunden, initialisiert den Drucker und wird beim Einschalten der Karte kurzzeitig aktiviert. **/ERROR** ist ein Eingang für Fehlermeldungen und kann z.b. mit **PAPER END** o.ä. Druckersignalen verbunden werden.
/STB gibt einen kurzen (625 ns) LOW-Impuls zur Datenübernahme aus. Wenn die GRIP-Karte für Spezialzwecke mit internen Taktfrequenzen über 20 MHz betrieben wird, sollte der **/STROBE**-Ausgang benutzt werden, der einen etwa 6 us langen Impuls erzeugt.

Über den CCON-Eingang können die Centronics-Datenleitungen für iregendwelche Sonderzwecke mit HIGH in den hochohmigen Zustand versetzt werden. Das muß die Position von J2 geändert werden:

J2 1-2,3-4: DATA1-8 immer aktiv.
2-3: DATA1-8 nur aktiv, wenn CCON auf LOW.

III.6.2. V24-Schnittstelle: Geht natürlich nur, wenn diese Schnittstelle nicht bereits vom Host benutzt wurde. Der Anschluß erfolgt wie unter III.5.3 beschrieben.

III.7 Anschluß der GRIP-COLOR

Für die Farb-Zusatzkarte GRIP-COLOR gibt es einen aus 32 Leitungen bestehenden **Grafikbus**. Dieser Grafikbus belegt die b-Leiste des Bussteckers, so daß die Erweiterungsplatinen wie normale Systemkarten in den ECB-Bus eingesteckt werden können. Natürlich braucht man dazu eine Busplatine, auf der auch die b-Reihen untereinander verbunden sind, z.b. **ECB-96** von CONITEC.

Wenn ein 96-poliger Bus nicht zur Verfügung steht, können die beiden Karten auch mit der 'Huckepack'-Methode verbunden werden. Dazu ist auf der Grundkarte nur ein 64-poliger Busstecker zu bestücken; die COLOR-Karte erhält gar keinen Busstecker. Die Verbindung erfolgt mit einer 32-poligen Pfostenreihe, die von unten in die freie b-Leiste der Grundkarte und von oben in die der COLOR-Karte eingelötet wird.

Beim Anschluß von GRIP-COLOR-Karten ist Jumper **J4** auf der GRIP-Grundkarte zu öffnen. Die Jumper auf den COLOR-Karten sind wie folgt zu stecken:

Erste COLOR-Karte: J1 Vorverbunden.
J2 Vorverbunden auf Position A (1-3).

Zweite COLOR-Karte: (wenn vorhanden)
J1 Aufgetrennt.
J2 Gesteckt auf Position B (2-4);
Vorverbindung auftrennen!

III.7.1: Kaskadierung von zwei COLOR-Karten: Die Karten sind mit einem Flachbandkabel über Stecker N3 verbunden. Nun werden die Farbton-Bits auf die beiden Karten verteilt:

Die erste Karte (Farbindex-Bits 0-2) soll z.b. für die oberen 2 Bits, die zweite (Farbindex-Bits 3-5) für die unteren 2 Bits jedes Rot-, Grün- und Blau-Intensitätswertes zuständig sein. Dazu sind auf beiden Karten die jeweils irrelevanten Ausgänge der Look-Up-Tafeln zu unterbrechen.
Auf der ersten Karte werden die Widerstände **R14,R15,R19,R20, R17,R9** entfernt, auf der zweiten **R24,R23,R30,R12,R28,R10**. Außerdem müssen auf der Karte, an die der Farbmonitor nicht angeschlossen wird, die D/A-Wandler entfernt werden (**P1-3, R11,13,16, 18,21,22,25,26,27,29,31,32**).

Bei der Programmierung der Farben (-> ESC ESC 'W' usw.) ist natürlich zu beachten, daß dann Farbindizes mit den gleichen Bits 0-2 zwangsläufig auch die gleichen Farbton-Bits 2-3 erhalten

(also etwa die gleiche Grundhelligkeit auf Rot, Grün und Blau haben), und ebenso Farbindizes mit den gleichen Bits 3-4 die gleichen Farbton-Bits 0-1.
Die völlige Freiheit bei der Farbzuordnung geht also bei Kaskadierung von zwei COLOR-Karten verloren. Dafür sind 128 verschiedene Farben gleichzeitig auf dem Bildschirm sichtbar.

III.7.2: Parallelschaltung zweier COLOR-Karten: Im Gegensatz zur Kaskadierung lassen sich hier zwei voneinander unabhängige Farbmonitore anschließen; dafür sind auf jedem Monitor natürlich wieder nur 16 Farben gleichzeitig sichtbar.
N3 auf den COLOR-Karten wird nicht benutzt, andere Umbauarbeiten entfallen ebenfalls. Bei der Farbprogrammierung zeichnen die Bits 0-2 des Farbindex auf den Monitor an der ersten, die Bits 3-5 auf den Monitor an der zweiten Karte.

III.7.3: Anschluß des Farbmonitors: Es kann ein Monitor mit RGB-Eingängen verwendet werden, am besten mit Analogeingängen. Bei RGB-Digitaleingängen kann der Monitor nur 8 Farben darstellen. Empfehlenswert, aber nur für Interlaced-Betrieb unbedingt erforderlich, ist ein leicht nachleuchtender Bildschirm.

Die Auflösung sollte wieder mindestens 18 MHz betragen. Optimal sind 800 Punkte horizontale Auflösung; bei weniger als 600 Punkten erhält man in Normalschrift ein so schlechtes Schriftbild, daß auf Breitschrift umgeschaltet werden muß.

Der Monitor ist an Stecker N2 anzuschließen; die Sync-Signale stehen in jeder gewünschten Polarität zur Verfügung:

```
/HS, /VS, HS, VS : direkt anschließen.
      /CSYNC : /HS,/VS verbinden und anschließen.
       CSYNC : /HS,/VS verbinden, an HS oder VS anschließen.
```

Die Farbtöne lassen sich über die Potis P1-3 einstellen und in der Amplitude an den Monitor anpassen. Der S/W-Monitor an der GRIP-Grundkarte kann angeschlossen bleiben und z.b. als reiner Textmonitor betrieben werden. In diesem Fall stehen allerdings nur 8 Farben für gleichzeitige Darstellung am Bildschirm zur Verfügung.

III.7.4: Anschluß eines monochromen Monitors: Hierbei wird die GRIP-COLOR nur zur Graustufendarstellung verwendet. Eine Farbe - z.b. das RED-Signal - wird zur Darstellung ausgewählt, GREEN und BLUE ignoriert (Z44 und Z43 brauchen dann auch nicht bestückt zu werden).

Bei einem geeigneten monochromen Monitor mit BAS-Eingang (z.b CRT-200 von CONITEC) werden /HS und /VS miteinander und über einen 68-Ohm-Widerstand mit RED verbunden; an der Verbindungsstelle kann dann der BAS-Eingang angeschlossen werden. R1 und R3 sind zu entfernen. Die Amplitude läßt sich über Poti P3 (für RED) einstellen; normalerweise ist das Poti auf etwa 1/4-Vollausschlag zu drehen.

III.8. Anschluß einer Kamera

Für manche Anwendungen will man ein externes Videosignal, z.b. von einer Kamera oder einem Videorecorder, mit der computererzeugten Grafik überlagern. Mit einfachem Mischen ist es nicht getan. Die Zeilen- und Bildsprünge beider Signale erfolgen dann nämlich noch völlig unsynchronisiert; das Ergebnis ist "Bildsalat".

Deshalb muß man das von GRIP erzeugte Videosignal mit der Zeilenfrequenz der Kamera synchronisieren. Zu diesem Zweck kann über die Leitung **CKPIX** ein externes Taktsignal eingespeist werden, wenn man die Brücke **J10** vorher öffnet. Die Frequenz sollte ungefähr 15 MHz betragen und läßt sich z.b. mit einer PLL-Schaltung aus dem Videosignal der Kamera gewinnen.

Diese Schaltung muß folgendes leisten:

- Synchronisation des CKPIX-Taktes mit HSYNC der Kamera
- Verlangsamung oder Stoppen des Taktes, solange HSYNC der GRIP früher kommt als HSYNC der Kamera
- Verlangsamung oder Stoppen des Taktes, solange VSYNC der GRIP früher kommt als VSYNC der Kamera
- Verlangsamung oder Stoppen des Taktes, solange im Interlaced-Betrieb das erste Halbbild der GRIP nicht mit dem ersten Halbbild der Kamera zusammenfällt.

Verlangsamung der Taktfrequenz ist besser als Stoppen, da alle Funktionen der GRIP - CPU-Takt oder RAM-Refresh - von diesem Takt abgeleitet werden. Eine Schaltung, die die Kriterien erfüllt, wird z.b. auf der **AVIP**-Videoprozessorkarte eingesetzt (-> AVIP-Handbuch von CONITEC).

Mit der gleichen Methode lassen sich übrigens auch mehrere GRIP-Karten zur Erzeugung eines gemeinsamen Videosignals kombinieren. Dies ist zur Erzeugung schneller, bewegter Echtzeit-Grafiken von Vorteil: Jede Karte ist dabei nur für ein einziges graphisches Objekt zuständig.

IV. GRIP-4-Befehlssatz

Das 32-KByte-EPROM auf der Karte enthält die Betriebs-Software. Aufgabe des Betriebsprogramms ist es, Daten vom Hostrechner zu empfangen, zu bearbeiten und dann entweder an eine der Schnittstellen auszugeben oder auf dem Bildschirm darzustellen.

Es gibt zwei verschiedene Darstellungsarten: Der **Non-Interlaced-Modus** arbeitet mit einer 8x8..8x10-Zeichenmatrix und 768x280 Bildschirmpunkten; der **Interlaced-Modus** bietet dagegen im Zeilensprungverfahren eine 8x16-Matrix und eine Auflösung von 768x560 Bildpunkten. Dafür sind Textausgabe und Vektorzeichnen um ca. 30% langsamer, und Druckerspooler bzw. zweite Bildschirmseite sind abgeschaltet.

Je nach Konfiguration der Host-Schnittstelle kann GRIP Befehle und Daten mit 8 oder nur mit 7 Bit Wortlänge empfangen. Es lassen sich alle Funktionen auch mit nur 7 Bit steuern; trotzdem wurden einige 8-Bit-Kommandozeichen definiert, um gegebenenfalls die Übertragung zu beschleunigen. Im ECB-Bus-Betrieb stehen in jedem Fall 8 Bit zur Verfügung.

Die Reaktion auf ein Befehls- oder Datenbyte richtet sich nach der **Emulation**, die die Karte gerade ausführt. Im **TVI-Modus** verhält sich GRIP wie ein TVI950-Terminal, im **Tektronix-Modus** wie ein Tektronix4010/4014-Grafikterminal mit Speicherbildröhre. Es sind nicht sämtliche, aber alle gebräuchlichen oder sinnvollen Befehle der beiden Terminals realisiert.

Die sogenannten **Doppelescape-Sequenzen** funktionieren in jedem Terminal-Modus. Sie werden durch zwei Escape-Zeichen (1Bh) eingeleitet und dienen zur Steuerung von Sonderfunktionen und zur Modi-Umschaltung.

Die meisten internen Datenströme zwischen asynchronen Prozessen laufen über FIFO-Zwischenspeicher (**Puffer**), um die Geschwindigkeit der Karte zu erhöhen und Datenverlust auszuschließen. Es gibt Puffer für Befehle (Host-Puffer), für die Daten von den drei anschließbaren Tastaturen (Tastaturpuffer), vom Quellenkanal und für den Soundgenerator (Soundpuffer). Der mit 32 KByte größte Zwischenspeicher ist der abschaltbare Puffer für die Ausgangsschnittstelle (Spooler).

Der Zustand des Tastaturpuffer, die Uhrzeit und einige Systemparameter werden in der **System-Statuszeile** am oberen Bildrand angezeigt. Die **User-Statuszeile** unten kann mit einem beliebigen Text beschrieben werden.

In der folgenden Kommandotabelle sind einige Befehle mit der

Bemerkung "def" (default) versehen. Die durch diese Befehle eingestellten Parameter sind beim Einschalten oder Rücksetzen der Karte ausgewählt. Die Farbbefehle sind nur in Verbindung mit dem Farbzusatz GRIP-COLOR wirksam.

Die Parameter haben folgende Bedeutung:

x,y,z = Binär-Byte (00h-FFh)
a,b,c = ASCII-Zeichen (20h-7Fh)
aa,bb = Hex-Byte (2 Zeichen), z.b. '41' (34h 31h) = 41h = 'A'
xx = Binäres 16-Bit-Wort (LSB zuerst), z.b. 0Ah 10h = 100Ah = 410
aaaa = Hex-Wort (4 Zeichen), z.b. '100A' (31h 30h 30h 41h) = 100Ah

ESC oder <ESC> = 1Bh
CR oder <CR> = 0Dh

Ein vorangestellter Schrägstrich (/) zeigt an, daß die folgenden Zeichen vom Host auf GRIP's Tastaturkanal empfangen werden können. Dabei ist das 8. Bit i.a. gesetzt, um die Zeichen von normalen Tastaturbytes zu unterscheiden.

Ein vorangestellter Pfeil (^) bedeutet ein Control-Zeichen, z.b. <^C> = 03h.

Zeichen in Anführungsstrichen (') bedeuten ASCII-Zeichen (z.b. 'A'), in Klammern Binär-Bytes (z.b. 41h). Bei einem Binär-Byte werden die Bits in der Form

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0

dargestellt, wenn sie einzeln unterschiedliche Bedeutung haben.

IV.1: Der TVI-Modus

Im TVI-Betrieb, der nach Reset automatisch aktiv ist, wird von der GRIP das Verhalten eines Televideo-950-Terminals simuliert. Natürlich ist diese Simulation nicht vollständig: Die verschiedenen Off-Line-Edit-Modi des TVI950, die sowieso niemand benutzt, fielen der Übersichtlichkeit des Befehlssatzes zum Opfer. Diverse hardware-abhängige Befehle wurden aus dem gleichen Grund nicht in den TVI-Modus übernommen, sondern mit Doppelescape-Sequenzen realisiert. Außerdem mußten viele neue Befehle implementiert werden, z.b. für Scrollen und Zeichenattribute, die beim TVI-Terminal nicht existieren.

Die Simulation erstreckt sich jedoch über alle Bildschirm-Editbefehle (Zeilenlöschen, -einfügen, Cursorpositionierung usw.), über einige Sonderbefehle wie z.b. Cursor-Umdefinition sowie über die Strichgrafik, die mit Sonderzeichen gemacht wird und z.b. von Spreadsheet-Programmen wie MULTIPLAN unterstützt wird. Damit lassen sich die Installationsmenues von Softwarepaketen wie WORDSTAR usw. bedenkenlos benutzen.

Zum Einschalten des TVI-Modus dienen die folgenden Doppelescape-Sequenzen:

ESC ESC 'A' TVI-Grundmodus einschalten.

Dies ist die Default-Einstellung. Alle Zeichen werden in die Grundebene geschrieben; mehrfarbige Darstellung geht nur in bereits vorher mit verschiedenen Farbindizes (s.u.) beschriebenen Bildschirmbereichen. Im Schwarzweißbetrieb ohne COLOR-Karte sollte immer diese Einstellung gewählt werden.

ESC ESC 'a' TVI-Farbmodus einschalten.

Alle Zeichen werden in dem eingestellten Farbindex (s.u.) geschrieben, der auch jederzeit gewechselt werden kann. In diesem Modus sind ohne Einschränkung mehrfarbige Texte möglich; dafür erfolgt die Zeichenübertragung deutlich langsamer, und Zeilen-Einfügen und -Löschen können zu Farbveränderungen führen.

IV.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. Bewirkt das gleiche 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.

III.1.2: TVI-Escape-Sequenzen

ESC '+' Löschen wie <^L>.
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
'5': User-Cursor blinkend
'6': User-Cursor stehend
'7': User-Cursor twinkling

Der 'User-Cursor' ist das Zeichen 00h (01h) im ASCII-Zeichensatz, er kann beliebig umdefiniert werden. Die Blinkfrequenz des Cursors läßt sich über die Systempatch-Adresse **4675h** verändern (Systempatch s.u.).

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, die unterste Zeile geht verloren. 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 !

s1-0 = 00: Index tief (Subscript)
01: Indexschrift
10: Index hoch (Superscript)
11: Normale Schrift (def)

b = Breitschrift
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. -> Kanal 4), 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.

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

Normalschrift: The Quick Brown Fox Jumps Over The Lazy Dog!

Indexschrift: The Quick Brown Fox Jumps Over The Lazy Dog!

Breitschrift: The Quick Brown Fox ..

Index breit: The Quick Brown Fox ...

Unterstrichen: The Quick Brown Fox Jumps Over The Lazy Dog!

Invertiert: The Quick Brown Fox Jumps Over The Lazy Dog!

Durchgestrichen: The Quick Brown Fox Jumps Over The Lazy Dog!

ESC ')' Invertieren ein.
ESC '(' Invertieren aus (def).

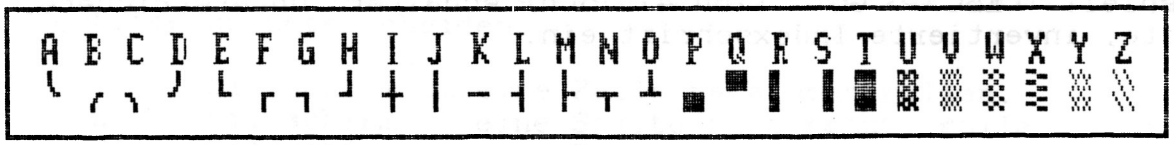
Diese Befehle lassen die übrigen Attribute unverändert und sollten für die Invertierungs-Option von WORDSTAR, TURBO-PASCAL usw. verwendet werden.

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. Das Dunkelschalten geht auch mit der Doppelescape-Sequenz <ESC> <ESC> '5' (s.u.).

ESC '$' Strichgrafik ein.
ESC '%' 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.



GRIP-Strichgrafik (MULTIPLAN-kompatibel)

IV.2: Der Tektronix-Modus

Während der TVI-Modus für die Textverarbeitung gedacht ist, dient der Tektronix-Modus der Grafikdarstellung. Er wird mit folgenden Doppelescape-Sequenzen eingeschaltet:

ESC ESC 'T' Tektronix-Grundmodus einschalten.

Vektoren und Zeichen werden in die Grundebene geschrieben. Dieser Modus muß eingeschaltet werden, wenn keine COLOR-Karte zur Verfügung steht.

ESC ESC 'X' Tektronix-Farbmodus einschalten (nur mit COLOR-Karte).

Vektoren und Zeichen werden in dem eingestellten Farbindex (s.u.) in die Farbebenen geschrieben. Wenn zwei Vektoren übereinandergeschrieben werden, überdeckt der Farbindex des neuen den des alten Vektors. Es wird empfohlen, diesen Modus zu verwenden, wenn eine COLOR-Karte zur Verfügung steht.

Das Tektronix-Terminal kann selbst in unterschiedlichen Modi arbeiten. Im **Alphamodus** verhält es sich wie ein "abgemagertes" Text-Terminal. Cursorpositionieren und Editieren sind nur eingeschränkt möglich, darum sollte zur Textverarbeitung der TVI-Modus verwendet werden.
Die Stärke des Tektronix-Terminals liegt in seiner Grafikfähigkeit. Im **Punkt- oder Vektormodus** zeichnet es Punkte bzw. Linien zwischen vorgegebenen Bildschirmkoordinaten. Im **Inkrementalmodus** wird ein mechanischer Plotter simuliert; es können kleinere Figuren aus Einzelpunkten schrittweise zusammengesetzt werden.

Es sind einige Befehle implementiert, die über den Original-Tektronix-Befehlssatz hinausgehen, z.b. Löschen, Invertieren und Flächenfüllen (ab Mitte 1986).
Wird im Tektronix-Modus ein Zeichen auf eine Stelle geschrieben, auf der bereits eins steht, so werden beide Zeichen übereinander dargestellt. Im TVI-Modus dagegen wird das "ältere" Zeichen gelöscht.

GRIP kann jederzeit mit den Doppelescape-Sequenzen zwischen TVI- und Tektronixmodus hin- und herschalten. Der Bildschirminhalt wird davon nicht beeinflußt.

IV.2.1: Tektronix-Control-Codes

^M (0Dh) Alphamodus einschalten. Der Alpha-Cursor wird auf den Beginn der Zeile 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 man nur kurz den Vektormodus verlassen hat und die Anfangskoordinate noch der letzten Koordinate entspricht.

^^ (1Eh) Inkrementalmodus einschalten.

^Q (11h) Lösche mit Schwarz (-> Hintergrundfarbe).
^R (12h) Zeichne invers (Schwarz <-> Weiß, Vordergrundfarbe).
^S (13h) Zeichne mit Weiß (-> Vordergrundfarbe, default).

Im Grundmodus sind diese drei Befehle klar - löschen, invertieren und zeichnen. Im Farbmodus erfolgt zeichnen und löschen in den mit ESC 'V' bzw. ESC 'v' (s.u.) eingestellten Vorder- und Hintergrundfarben. Das Invertieren dagegen invertiert nur das 7. Bit des Farbindex, d.h. das Bit der Grundebene; in die Farbebenen wird dabei mit der Vordergrundfarbe gezeichnet.

IV.2.2: Tektronix-Escapesequenzen

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

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.

Die zuletzt eingestellte Grafikkoordinate rollt mit, die Position relativ zum Nullpunkt verändert sich also.

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

ESC '`' Vektortyp: durchgehende Linie (default).
ESC 'a' Vektortyp: gepunktete Linie.
ESC 'b' Vektortyp: Strichpunkt-Linie.
ESC 'c' Vektortyp: kurz gestrichelte Linie.
ESC 'd' Vektortyp: lang gestrichelte Linie.

ESC 'F' Fülle Fläche (ab Mitte 1986).

Der Grafik-Cursor muß zum Flächenfüllen in einem vollständig umrandeten Gebiet stehen. Die Umrandung wird in der mit <ESC> <ESC> 'v' (s.u.) eingestellten Bit-Ebene erkannt. Gefüllt wird in der aktuellen Vektor-Farbe mit einem Muster, das dem durch **<DEL> (7Fh)** codierten Zeichen des aktuellen Zeichensatzes entspricht und sich alle 8x8 (bzw. 8x16) Rasterpunkte wiederholt. Dieses Zeichen ist normalerweise eine Graufläche und kann durch Umprogrammieren des Zeichensatzes geändert werden.

IV.2.3: Der 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 **768x560 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 beitra-

gen. 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 gemein hat - 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 SETXY (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 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' bestimmt werden, die Farbe mit <ESC> 'V', die Blinkfarbe mit <ESC> 'Y' (s.u.).

Zwei aufeinanderfolgende gleiche Koordinaten erzeugen nur einen Punkt. Wir erinnern uns: 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 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 (<^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 nämlich nach einem Algorithmus berechnet, der rückwärts manchmal die "Stufen" anderswohin setzt als vorwärts.

IV.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.

IV.2.5: Inkrementalmodus

Im Inkrementalmodus wird von der letzten Grafikkoordinate 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.
                              'I' (49h): diagonal nach rechts unten.
   'B'    *    'A'            'H' (48h): nach unten.
                              'J' (4Ah): diagonal nach links unten.
                              'B' (42h): nach links.
   'J'   'H'   'I'            '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).
```

IV.3: Farbe und Blinken

Vorab eine kurze Erläuterung der Termini.

GRIP's Farbgrafik funktioniert 'zweistufig': In das durch die Farbebenen erweiterte Video-RAM wird beim Schreiben an jede Pixelposition ein 7-Bit-**Farbindex** eingetragen. Bit 6 entspricht dabei dem Wert, der in der Grundebene steht, die Bits 0-2 stehen in den drei Farbebenen der ersten Farbkarte, Bit 3-5 in der zweiten Farbkarte (falls vorhanden).

Jedes Pixel hat also seinen Farbindex, der zunächst nur ein 7-Bit-Wert ist und noch nichts mit sichtbaren Farben zu tun hat. Die Zuordnung geschieht in der "zweiten Stufe": Jeder der insgesamt 128 Indizes bekommt über die Look-Up-Tafel (s.o.) einen **Farbton** zugewiesen, der additiv aus Rot-, Grün- und Blauanteil zusammengemixt wird. Da sich jeder Anteil in 16 Stufen einstellen läßt, sind insgesamt 16x16x16=4096 Farbtöne möglich, von denen natürlich nur die 128 zugewiesenen am Bildschirm sichtbar sind.

Da sich jederzeit ein Farbton umdefinieren läßt, ist es kein Problem, z.b. alle bisher roten Linien plötzlich grün zu färben. Am Inhalt des Video-RAM's, d.h. am Farbindex, ändert sich dabei nichts; nur ein Wert in der Look-Up-Tafel wird umgeschrieben.

Die Werte in den Farbebenen, d.h. Bit 0-5 des Farbindex, lassen sich nur im Farbmodus (-> ESC ESC 'a', ESC ESC 'X') verändern. Der Grundmodus schreibt nur in die Grundebene, beeinflußt also nur Bit 6 des Farbindex.

Die folgenden Befehle sind sowohl im TVI- wie auch im Tektronix-Farbmodus in gleicher Weise wirksam:

ESC 'W' x Vordergrundfarbe einstellen.
ESC 'w' x Hintergrundfarbe einstellen.

x = ! 0 ! FG ! FC5 ! FC4 ! FC3 ! FC2 ! FC1 ! FC0 !

FG = Farbindex-Bit Grundebene
FC3-5 = Farbindex-Bits zweite COLOR-Karte
FC0-2 = Farbindex-Bits erste COLOR-Karte

ESC 'V' a Vordergrundfarbe (transparent) einstellen.
ESC 'v' a Hintergrundfarbe (transparent) einstellen.

a = Farbindex (FC0-5) + 20h

In dem gewählten Farbindex (FG=Grundebene, FC=Farbebenen)

werden von nun an Vektoren und Zeichen im Farbmodus geschrieben. Die Bits FC3-5 sind nur für eine zweite COLOR-Karte relevant.

Der **Vordergrund-Farbindex** gilt im Tektronix-Farbmodus für das Zeichnen (-> ^S), im TVI-Farbmodus wird darin das Textzeichen geschrieben. Der **Hintergrund-Farbindex** gilt im Tektronix-Farbmodus für das Löschen (-> ^Q), im TVI-Farbmodus für den Zeichenhintergrund und in beiden Fällen für das Bildschirmlöschen. Der Tektronix-Befehl ^R ignoriert FG, invertiert nur das Bit in der Grundebene und schreibt den Vordergrund-Farbindex (FC0-5) in die Farbebenen.

Ein transparenter Farbindex (nach <ESC> 'V','v') schreibt nur in die Farbebenen, jedoch nicht mehr in die Grundebene. Dies ist z.b. für den Betrieb mit zwei getrennten Monitoren für Text und Farbgrafik sinnvoll. Außerdem erfolgt das Zeichnen eines transparenten Farbindex schneller.

Defaultmäßig ist der Farbindex 0000001b (01h) als Vordergrundfarbe und 0000000b (00h) als Hintergrundfarbe eingestellt.

ESC ESC 'v' a Bildspeicher-Ebene einstellen zum direkten Einschreiben, Auslesen oder Ausdrucken.

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

Dieser Befehl ist notwendig für -> Hardcopy, Flächenfüllen oder zum direkten Zugriff auf das Video-RAM (-> Kanal 3). Der Parameter a bezieht sich auf das gewählte Bit (FG,FC0-5) des Farbindex.

ESC ESC 'W' a r g b	Farbton zuordnen (für Indizes mit FG=0).
ESC ESC 'w' a r g b	Farbton zuordnen (für Indizes mit FG=1).
ESC ESC 'W' ' '	Default-Farbtöne einstellen.
ESC ESC 'Y' a r g b	Blink-Farbton zuordnen (Indizes mit FG=0).
ESC ESC 'y' a r g b	Blink-Farbton zuordnen (Indizes mit FG=1).
ESC ESC 'Y' ' '	Blinken abschalten (default).

a = Farbindex (FC0-5) + 20h
r = Rot-Intensität (0..15) + 30h
g = Grün-Intensität (0..15) + 30h
b = Blau-Intensität (0..15) + 30h

Die am Bildschirm sichtbare Farbe, die einem bestimmten Farbindex zugeordnet ist, wird mit <ESC> <ESC> 'w' bzw. 'W' eingestellt.

Bei jeder Farbe kann der Farbton und die Helligkeit unabhängig über die Rot-, Grün- und Blau-Intensität in je 16 Stufen bestimmt werden. '0' stellt z.b. die niedrigste, '8' eine mittlere, '?' die höchste Intensität ein. Durch die Mischung der drei Farbanteile ergeben sich insgesamt 16x16x16=4096 einstellbare Farbtöne.

Im Betrieb mit getrenntem Text- und (Farb-)Grafikmonitor sollten für FG=0 und FG=1 die gleichen Farbtöne eingestellt werden, damit die Grundebene auf dem Farbmonitor nicht sichtbar ist.

Eine mit <ESC> <ESC> 'V'/'v' zugeordnete Farbe steht normalerweise stetig am Bildschirm. Man kann den entsprechenden Farbindex jedoch auch zwischen zwei beliebigen Farbtönen **blinken** lassen. Dazu wird mit <ESC> <ESC> 'Y'/'y' eine **Blinkfarbe** festgelegt, die von der "Normalfarbe" verschieden sein muß. <ESC> <ESC> 'W'/'w' stellt Normal- und Blinkfarbe auf den gleichen Wert und schaltet damit das Blinken für den gewählten Farbindex ab. <ESC> <ESC> 'Y ' (1Bh 1Bh 59h 20h) schaltet das Blinken für alle Indizes ab.

Schnelle **Bewegungseffekte** lassen sich erzielen, wenn Teile des Bildes mit verschiedenen Farbindizes gezeichnet und dann rasch nacheinander sichtbar (durch Zuordnen eines Farbtons) und unsichtbar (durch Zuordnen von Schwarz bzw. des Hintergrund-Farbtons) gemacht werden.

Defaultmäßig ist der Vordergrund-Farbindex (01h) auf rot (r='?',g=b='0') und der Hintergrund-Farbindex (00h) auf schwarz (r=g=b='0') eingestellt. Darum erscheint beim Einschalten des Farbmodus auf dem Farbmonitor rote Schrift auf schwarzem Grund.

Beispiel: <ESC> <ESC> 'W 500' <ESC> <ESC> 'w >>0' <ESC> <ESC> 'A' (1Bh 1Bh 57h 20h 35h 30h 30h 1Bh 1Bh 77h 20h 3Eh 3Eh 30h 1Bh 1Bh 41 schaltet den TVI-Grundmodus mit knallgelbem Text auf dunkelrotem Hintergrund ein.

IV.4: Die Tastatur

Die beiden folgenden Befehle dienen der Anpassung und Auswahl der Tastatur:

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

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

w = 0: Tastatur-Wortlänge 8 Bit
1: Tastatur-Wortlänge 7 Bit

i = 0: Tastatur direkt (def)
1: Tastatur invertiert

c1-0 = 00: Parallele Tastatur
01: IBM-Tastatur
10: Serielle Tastatur (V24 oder TTL)
11: Tastatur abgeschaltet

h = 0: Host-Wortlänge 8 Bit
1: Host-Wortlänge 7 Bit (de

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

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

a = '0': Baudrate wie J3b (def)
'1': 50 Baud
'2': 75 Baud
'3': 110 Baud
'4': 150 Baud
'5': 300 Baud
'6': 600 Baud
'7': 1200 Baud
'8': 2400 Baud
'9': 4800 Baud

Der Default-Tastaturkanal wird über Jumper J3b (-> III.3) eingestellt. Jede der drei Tastaturkanäle ist mit 64 Byte FIFO-Zwischenspeicher (**Tastaturpuffer**) versehen.

Wenn auf einem der drei Tastaturkanäle ein Tastencode gesendet wird, gelangt er zunächst in den zugeordneten Tastaturpuffer. Ist der Tastaturkanal aktiv, d.h. mit c0-1 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 (-> Quellenkanal).

Die Default-Polarität der seriellen Tastatur wird beim Einschalten anhand des Ruhepegels automatisch erkannt und eingestellt.
Das Abschalten aller drei Tastaturen (C1-0 auf 00) ist sinnvoll, wenn von GRIP's **Quellenkanal** ein längerer Datenstrom erwartet wird, z.b. beim direkten Auslesen des Video-RAM's.

ACHTUNG: Der Befehl <ESC> <ESC> '4' ist nicht voll kompatibel zu den älteren GRIP-2/3-Versionen!
ACHTUNG: Wenn Sie einen Akku (2-3 Volt) auf der UBAT-Leitung des ECB-Bus (a24) angeschlossen haben, bleibt die Einstellung nach <ESC> <ESC> '4' auch beim Ausschalten der Spannung erhalten! Sie wird erst dann wieder gelöscht, wenn Sie die Karte aus dem Bus ziehen.

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

IV.5: Die Kanäle

Unter diesem Begriff sind Funktionsgruppen der GRIP zusammengefaßt, an die der Hostrechner Daten senden kann (-> **Input-Kanäle**), wie z.b. das Druckerinterface oder eine der internen Tabellen. Ebenso gibt es Kanäle, von denen der Hostrechner Daten empfangen kann (-> **Output-Kanäle**), etwa die Tastaturkanäle oder das Video-RAM.

Zwei der Inputkanäle haben wir bereits kennengelernt, die **Text-Konsole** (TVI-Modus) und die **Grafik-Konsole** (Tektronix-Modus). Vom Host gesendete Bytes 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 beiden Konsolenkanälen und den drei Tastaturkanälen kennt GRIP noch zwölf **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 auf 0 (wie bei den normalen ASCII-Zeichen 20h..7Fh), geht das Byte zu einer der Konsolen; ist es auf 1, wird es zum gerade aktiven Zielkanal gesendet. Dabei wird das 8. Bit vor der Übertragung intern 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.

IV.5.1: Kanalwahl und Transfer

Folgende Doppelescape-Sequenzen wählen einen der Kanäle aus:

ESC ESC 'C' a Zielkanal wählen.

a =	'0':	Centronics (def)	'8':	Video-RAM Farbe
	'1':	ECB-Bus	'9':	User-Statuszeile
	'2':	V24-Schnittstelle	'A':	Systempatch
	'3':	Video-RAM	'B':	Druckerpatch
	'4':	Zeichensätze		
	'5':	Userprogramm		
	'6':	Tasten-Umcodetabelle		
	'7':	Soundgenerator		

ESC ESC 'c' a Quellenkanal wählen.

a =	'0': Quellenkanal aus	'8':	Video-RAM Farbe
	'1': ECB-Bus	'9':	User-Statuszeile
	'2': V24-Schnittstelle	'A':	Systempatch
	'3': Video-RAM	'B':	Druckerpatch
	'4': Zeichensätze	'C':	Paralleltastatur
	'5': Userprogramm	'D':	Serielle Tastatur
	'6': Tasten-Umcodetabelle	'E':	IBM-Tastatur
	'7': Soundgenerator		

Der Default-Quellenkanal ist die V24-Schnittstelle, falls GRIP vom ECB-Bus aus betrieben wird (J3a offen); ansonsten der ECB-Bus. Die Tastaturen können nur dann als Quellenkanal benutzt werden, wenn sie nicht bereits als Tastaturkanal definiert sind. Ziel- oder Quellenkanal dürfen mit dem Host-Kanal identisch sein; damit kann man z.b. während einerHardcopy (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 ungeändert zum Zielkanal.

ESC ESC 'F' aa..<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 gelöschtem 8. Bit direkt zum Zielkanal senden. Dieser Modus ist **immer** eingestellt und stellt die einfachste Art der Übertragung zum Zielkanal dar. Er sollte, wenn möglich, für die Implementation des CP/M-LST:-Device verwendet werden.

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> 'F' '2B3132330D' <CR>.
(1Bh 1Bh 46h 32h 42h 33h 31h 33h 32h 33h 33h 30h 44h 0Dh).

String: <ESC> <ESC> 'f' 05h 00h '+123' <CR>.
(1Bh 1Bh 66h 05h 00h 2Bh 31h 32h 33h 0Dh).

Ähnliche Befehle gibt es 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 V24), 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> 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 (1-65535). 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.. Bytes vom Quellenkanal holen. xx gibt die Anzahl an und kann den Wert von 1-65535 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. Man sollte aber nach Ende der Übertragung immer auf den Default-Kanal zurückschalten.

Das Lesen eines Bytes vom Quellenkanal, z.b. von der V24-Schnittstelle, erfordert insgesamt folgende Schritte:

1. Quellenkanal wählen, z.b. <ESC> <ESC> 'c2' für V24.

2. Hex-Abfragebefehl geben, z.b. <ESC> <ESC> 'E'.

3. GRIP abfragen, bis ein Byte das 8. Bit gesetzt hat. Voraussetzung: h-Bit des Tastaturkanals (-> IV.4) 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 6 zurückgehen.

Im Abschnitt 'V24-Schnittstelle' (s.u.) ist das Ganze in Z80-Assemblersprache realisiert. Die einzelnen Kanäle werden in den nächsten Abschnitten noch näher erläutert.

IV.5.2 Centronics-Schnittstelle (Zielkanal 0)

An diese Schnittstelle wird normalerweise ein Drucker angeschlossen. Die Datenübertragung sollte der Einfachheit halber im Direktmodus erfolgen. Es wird empfohlen, bei der Implementierung für CP/M den LST:-Device defaultmäßig der Centronics-Schnittstelle zuzuordnen und den Zielkanal nach irgendwelchen anderen Übertragungen immer wieder auf Kanal 0 zurückzustellen.

Natürlich ist es auch möglich, die Centronics-Schnittstelle als **8-Bit-Universalport** für irgendwelche Steuerzwecke zu benutzen.

Der Centronics-Schnittstelle kann ein **32-Kbyte-Spooler** vorgeschaltet werden (-> <ESC> <ESC> '5'), der etwa 15 Seiten Text faßt.

Zur Implementierung als LST:-Device im CP/M-BIOS werden folgende Routinen (Z80-Code) vorgeschlagen (GRIPOUT,GRIPOST -> Kapitel III):

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

```
CENOUT: SET  7,C        ;Direktmodus: Achtes Bit auf 1
        JP   GRIPOUT    ;Ausgabe an Zielkanal (-> Kapitel III)
```

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    ;<ESC>
        CALL GRIPOUT    ;<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 (-> ESC ESC 'L') abgefragt werden.

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

Hier wird normalerweise der Host angeschlossen. Für Sonderzwecke, z.b. zur **Kopplung zweier Rechner**, kann es jedoch sinnvoll sein, den Host über V24 anzuschließen und die ECB-Bus-Schnittstelle als normalen Übertragungskanal zu einem zweiten Rechner zu benutzen. Die Schnittstelle ist als Quellenkanal mit 256 Byte FIFO-Speicher gepuffert. Zur Installation siehe -> Kapitel III.

IV.5.4 Die V24-Schnittstelle (Ziel/Quellenkanal 2)

Wenn die V24/RS232-Schnittstelle nicht vom Host belegt wird, kann sie unter CP/M als AUX:-Device oder als LST:-Device implementiert werden. Sender und Empfänger lassen sich mit verschiedenen Geschwindigkeiten betreiben, was für den eventuellen Anschluß eines Bildschirmtext-Modems wichtig ist. Zur Installation siehe -> Kapitel III.

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

ESC ESC '0' a Spezifiziere V24-Empfängerbaudrate.
ESC ESC '1' a Spezifiziere V24-Senderbaudrate.

a =		
'0':	Baudrate wie J3 (def)	'6': 600 Baud
'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	';': 30000 Baud

Statt der intern erzeugten Baudraten kann aber auch ein externer Baudratentakt für die V24-Schnittstelle über die Leitung BAUD (Jumper J6 -> Kap. III) eingespeist werden.

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

ESC ESC '2' x Definiere V24-Parameter.

x = ! 0 ! w1-0 ! s-0 ! p1-0 ! q !

w1-0 =		p1-0 =	
00:	Wortlänge 8 Bit (def)	01:	Parität aus (def)
01:	7 Bit	10:	Ungerade Parität
10:	6 Bit	11:	Gerade Parität
11:	5 Bit		

s1-0 =		q =	
01:	1 Stopbit (def)	0:	XON/XOFF aus (def)
10:	1.5 Stopbits	1:	XON/XOFF-Protokoll
11:	2 Stopbits		

XON/XOFF-Protokoll ist nur bei bidirektionalem Anschluß möglich. Dabei sendet die Schnittstelle ein XOFF-Zeichen (13h), wenn während des Datenempfangs der FIFO-Empfangspuffer überzulaufen droht (Platz nur noch für weniger als 16 Bytes). Daraufhin muß die angeschlossene Einheit das Senden unterbrechen. Ist der Puffer wieder leer, wird mit einem XON-Zeichen (11h) zum Weitersenden aufgefordert.

Ist der Host über die V24-Schnittstelle angeschlossen, muß bei Baudraten ab 30000 Baud oder bei Grafik-Betrieb (Flächenfüllen, Vektorzeichnen) mit RTS/CTS-Handshake oder mit XON/XOFF-Protokoll gearbeitet werden. XON/XOFF-Betrieb ist dann erforderlich, wenn der Hostrechner keine Handshake-Leitungen besitzt.

Beispiel: <ESC> <ESC> '2' <^T> (1Bh 1Bh 32h 14h) setzt 8 Bit Wortlänge, 1.5 Stoppbits, ungerade Parität und kein XON/XOFF-Protokoll.

Der V24-Empfänger enthält ebenso wie das ECB-Bus-Interface 256 Byte FIFO-Datenspeicher, der auch dann Daten aufnimmt, wenn Kanal 2 nicht eingeschaltet ist. Dem Sender läßt sich, wie der Centronics-Schnittstelle, der **Spooler** zuschalten. Das ist sinnvoll, wenn z.b. der LST:-Device über diese Schnittstelle implementiert wird.

Die folgenden Routinen 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   ;siehe Kapitel III
         CALL GRIPOUT   ;Zweimal ESC
         POP  BC        ;Befehlszeichen zurück
         JP   GRIPOUT   ;Zeichen ausgeben
```

SEROUT sendet das Zeichen in Register C an die V24-Schnittstelle.

```
SEROUT:  PUSH BC        ;Zeichen retten
         LD   C,43H     ;'C' für Zielkanal
         CALL DESCOUT   ;<ESC> <ESC> 'C' ausgeben
         LD   C,32H     ;'2' für V24
         CALL GRIPOUT   ;Zielkanal auf V24

         POP  BC        ;Zeichen zurück
         SET  7,C       ;Achtes Bit setzen für Direktmodus
         CALL GRIPOUT   ;Zeichen senden an V24

         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   ;V24-Sender ist ready, wenn GRIP ready
```

Anders verhält es sich mit **SERIST**, die den **Empfängerstatus** abfragt (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 V24
        CALL GRIPOUT    ;Quellenkanal auf V24

        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 Ihre 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          ;
        DEC  A          ;FFh = Ready-Zeichen
        RET

NIB2    DS   1          ;Speicherplatz für
NIB1    DS   1          ;V24-Daten (ASCII-Hex)
```

SERIN holt ein Zeichen von der V24-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) konvertieren
```

IV.5.5 Direkter Video-RAM-Transfer (Ziel/Quellenkanäle 3 und 8)

Dieser Befehl 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. Farb-RAM) für solche Zwecke zur Verfügung.

Der Bildschirmspeicher 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.

Die einzelnen Bytes bilden von oben nach unten die Pixelzeilen des Blocks. Ein Auschnitt 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 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 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 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 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 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 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 x N+15  x x x x x N+23  x x x x x N+31 . . .
. . . N+768 x 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 x N+777 x x x x x N+785 x x x x x N+793 . . .
        .             .             .             .
        .             .             .             .
        .             .             .             .
```

Bei Kanal 3 wird die eingestellte Vordergrundfarbe eingeschrieben. Ausgelesen wird die Bit-Ebene, die durch <ESC> <ESC> 'v' (s.o.) bestimmt wird.

Kanal 8 überträgt in einem besonderen Format die Farbindizes in die Farbebenen der ersten COLOR-Karte. Für jede Byte-Position im Bildschirmspeicher werden dabei vier Bytes übertragen:

IDX0 - IDX1 - IDX2 - IDXG

IDX0-2 enthalten die Informationen für die Farbindex-Bits 0-3 der jeweiligen Bit-Pixel-Positionen, **IDXG** für die Farbindex-Bits 7 (Grundebene). Die Bytes IDX0-2 werden also direkt in die Farbebenen 0-2 geschrieben und IDXG in die Grundebene. Die Anzahl der übertragenen Bytes auf Kanal 8 muß immer durch 4 teilbar sein!

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

ESC ESC 'I' aaaa Setze Video-RAM-Adresse im ASCII-Hex-Format.
ESC ESC 'i' xx Setze Video-RAM-Adresse (binär).

Dieser Befehl setzt zugleich den Farbebenenzähler für die Farbbildübertragung zurück. Die eingestellte Adresse wird nach jedem übertragenen Byte 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.

Beispiel: <ESC> <ESC> 'I4000' (1Bh 1Bh 49h 34h 30h 30h 30h) setzt die Adresse auf 4000h (21. Blockzeile, 32. Block von links, erstes Byte).

IV.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.

Es gibt 18 vordefinierte Zeichensätze: Acht ASCII-Zeichensätze, ein Spezialzeichensatz mit dem griechischen Alphabet und mathematischen Sondersymbolen, und verkleinerte Versionen aller dieser Zeichensätze für die Indexschrift.

Zwischen Standard- und Indexzeichensatz kann man mit <ESC> 'G' (s.o.) umschalten. 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 acht ASCII-Zeichensätze '0'-'7' überlappen sich, d.h. sie haben einen Großteil der Zeichen gemeinsam. Zeichensatz '8' ist völlig verschieden. Der Parameter '9' initialisiert alle Zeichensätze aus dem EPROM, ':' - '=' nur einige.

ASCII

Britisch

Deutsch

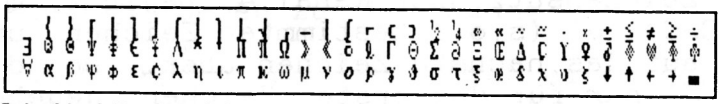
Dänisch

Französisch

Schwedisch

Italienisch

Spanisch



Griechisch/Sonderzeichen

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

Im Interlaced-Modus passen wegen der doppelten Zeichenmatrix nur halb so viele Zeichen in das RAM. Der Befehl <ESC> <ESC> '78' überschreibt darum dort die ASCII-Zeichensätze.

Jedes Zeichen des gerade aktiven Zeichensatzes kann vom Benutzer umdefiniert werden. Der folgende Befehl wählt das Zeichen an:

ESC ESC 'J' a Setze zu übertragendes Zeichen im aktuellen Zeichensatz.

Nach jeweils 8 (non-interlaced) bzw. 16 (interlaced) über Kanal 4 übertragenen Bytes ist das Zeichen fertig, und es geht weiter mit dem darauffolgenden Zeichen. Einige Zeichen in den ASCII-Zeichensätzen haben Sonderbedeutungen:

a = 00h: Cursorzeichen für User-Cursor (-> ESC '.')
01h: Zweites Cursorzeichen bei twinkling Cursor.
20h-7Eh: Normale Zeichen.
7Fh: Muster für Flächenfüllen.

Jedes Zeichen wird durch 8 Bytes definiert, die einen 8x8-Block bilden (interlaced: 16 Bytes in einem 8x16-Block). Die Bytes sind die Zeilen des Blocks, der oben links mit Bit 0 von Byte 1 beginnt. Ein auf 1 gesetztes Bit erscheint am Bildschirm als heller Punkt.

Beispiel: Zeichen '1' Bit 01234567 (non-interlaced)

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)

Die Übertragung in den Zeichensatz über Kanal 4 beginnt mit Byte 0 des durch <ESC> <ESC> 'J' (s.o.) bestimmten Zeichen und geht alle 8 bzw. 16 Bytes auf das nächste Zeichen über, so daß der Befehl <ESC> <ESC> 'J' nur für das erste Zeichen gegeben werden muß.

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

Zunächst non-interlaced 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 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.

IV.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 20 Ebenen 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. 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 Organisation der I/O-Ports und die Programmierung des Grafik-Controllers ist im Anhang beschrieben.

Unabhängig vom Userprogramm gibt es noch eine einfachere Möglichkeit, direkt auf den Controller oder andere 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. y wird an den Port x ausgegeben.

ACHTUNG: Durch ein fehlerhaftes Userprogramm oder direkte Portausgabe kann man GRIP zum Aussteigen bringen! Diese Funktionen sind nur etwas für Profis.

IV.5.8 Tasten-Umcodetabelle (Ziel/Quellenkanal 6)

Für jede gedrückte Taste kann statt des Original-Tastencodes ein beliebiger Zeichen-String zum Host gesendet werden. Auf diese Weise lassen sich Tasten umdefinieren oder Funktionstasten mit irgendwelche 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 (1 Byte) - String - nächster Tastencode usw. . . . . 00h 00h.

Bei Stringlänge 0 wird die entsprechende Taste einfach ausgeblendet, d.h. kein String gesendet. Zwei aufeinanderfolgende Nullen 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 nicht umcodierten Originalcodes!

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 unter CP/M 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 = Umcodetabelle)
ESC ESC 'F'                     (Hex-Übertragung einleiten)
'59015A5A0159'                  (Y mit Z vertauschen)
'79017A7A0179'                  (y mit z vertauschen)
'19011A1A0119'                  (^Y mit ^Z vertauschen)
'880444495200'                  (88h sendet DIR <CR>)
'0000'                          (Ende der Tabelle)
<CR>                            (Ende der Hex-Übertragung)
ESC ESC 'C0'                    (Zielkanal wieder Centronics)
```

IV.5.9 Der Soundgenerator (Ziel/Quellenkanal 7)

Der Soundgenerator kann eine Folge von Tönen mit wählbarer Frequenz, Dauer und Laufstärke 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 kHz)/(b1)   für gerade b1
F = (9.375 kHz)/(b1)   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 Lautstärke und Dauer des Tons. Es ist folgendermaßen aufgebaut:

```
b2 = ! 0 ! 11-10 ! d4-d0 !

11-0: 00 = Pause
      01 = Leiser Ton
      10 = Mittlerer Ton
      11 = Lauter 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, die von Bernhard Emese, dem Autor der GRIP-Software, eigens für GRIP komponiert wurde:

```
ESC ESC 'C7'                    (Zielkanal = Soundgenerator)
ESC ESC 'F'                     (Hex-Übertragung einleiten)
'42305824580458245030 58'       (Jetzt kommen die Töne)
'22681D4622460D4222420D'
'0000'                          (Ende der Melodie)
<CR>                            (Ende der Übertragung)
ESC ESC 'C0'                    (Zielkanal wieder Centronics)
```

IV.5.10 Die Statuszeilen (Ziel/Quellenkanal 9)

GRIP 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.

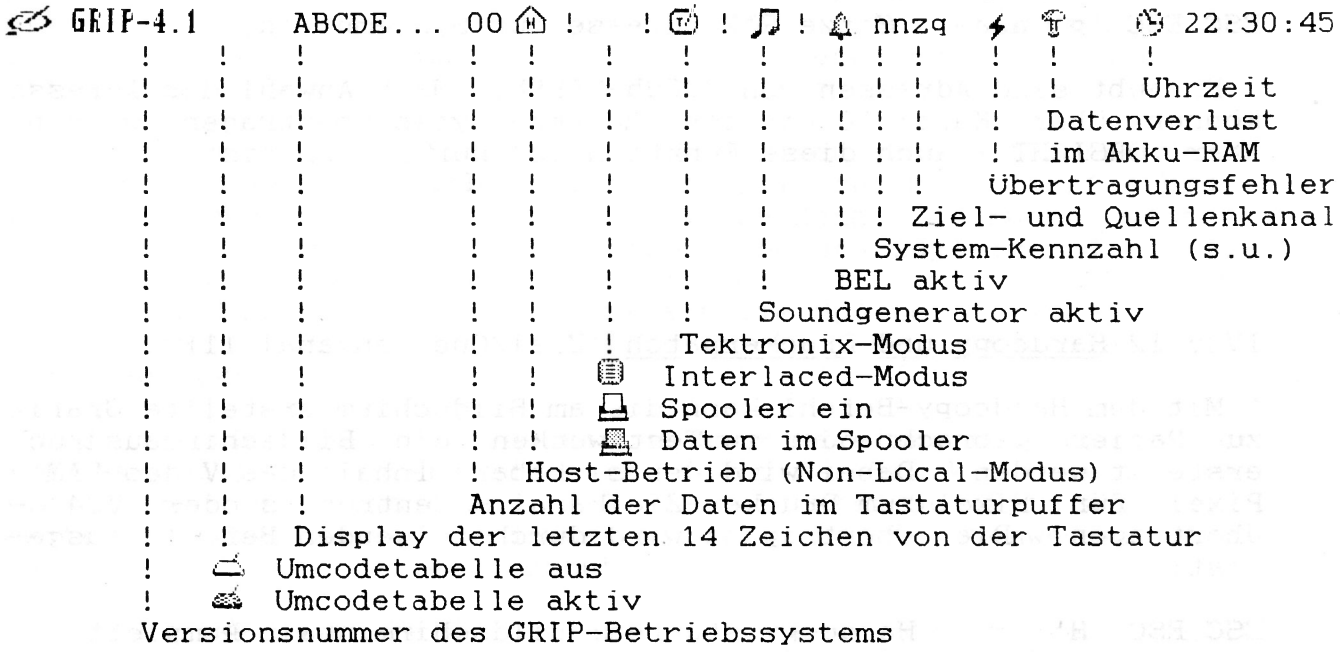
Die Statuszeilen lassen sich mit folgendem Befehl ab- oder umschalten:

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.

Zu beachten ist, daß für jede der Statuszeilen zwei Textzeilen verlorengehen. Nach Umschalten der Statuszeilen muß eventuell das Textformat neu gewählt werden! (-> <ESC> <ESC> '8').

Die Symbole der System-Statuszeile haben folgende Bedeutung:



Systemkennzahl nn: ! c ! x ! i ! k1-0 ! u ! h1-0 !

c = Farbmodus ein
x = XON/XOFF-Protokoll ein
i = Tastatur invers
k1-0 = 00: Parallel-Tastatur
01: IBM-Tastatur
10: Serielle Tastatur
11: Tastaturkanal aus
u = Umcodetabelle aktiv
h1-0 = 00: Local-Modus
01: Host = ECB-Bus
10: Host = V24

IV.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, z.b. Blinkfrequenz des Cursors und der Farben auf Adresse 4675h oder die Linienmaske für Vektoren auf Adresse 464Eh.

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

stellt:

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. Aber VORSICHT - auch diese Funktion ist nur für Profis!

IV.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 V24 - übertragen. Die Übertragung wird durch folgenden Befehl ausgelöst:

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

Dieser Befehl wir ignoriert, wenn ein anderer Zielkanal als Centronics, V24 oder ECB-Bus eingestellt ist. Bei dem Drucker muß es sich um einen grafikfähigen Matrixdrucker (ein- oder mehrfarbig) handeln. Für Farbdrucker 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/EPSON FX-80 oder kompatiblen.
'1': Gesamt-Hardcopy auf STAR (10X) oder kompatiblen.
'2': Gesamt-Hardcopy auf Siemens PT-88.

Es lassen sich beliebige Matrix-Drucker mit 8 oder mehr Nadeln für eine Hardcopy anpassen. Dazu muß mit Hilfe des 'Druckerpatch'-Kanals ein RAM-Feld programmiert werden, das aus einem **Steuerbyte** sowie einem **Initialisierungsstring (IS)** besteht. Durch den <ESC> <ESC> 'h'-Befehl kann das Feld für drei häufig verwendete Druckertypen bereits vordefiniert werden.

Bei der **Gesamt-Hardcopy** sendet GRIP den IS zum Zielkanal, danach folgen 560 Bytes, die die Pixelinformation (Text und Grafik) der linken vertikalen Randspalte von 8 Pixel Breite

enthalten. Das Ganze wird insgesamt 96mal wiederholt, wobei Spalte für Spalte ausgegeben wird. Auf dem Drucker erscheint der Bildschirminhalt also um 90 Grad gedreht. Bei Verwendung der COLOR-Karte kann die auszudruckende Bit-Ebene des Bildschirmspeichers mit <ESC> <ESC> 'v' vorgewählt werden.

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 Nadelbreiten und einen Wagenrücklauf auslösen und anschließend den Drucker so auf Grafikausgabe initialisieren, daß die nächsten 560 Byte als Bitmuster in einer Zeile abgedruckt werden. Bei spaltenweiser Hardcopy entfällt der Zeilenvorschub.

Das Druckerpatch-Feld beginnt mit einem Steuerbyte, danach folgt der IS. Das Steuerbyte hat folgenden Aufbau:

! s ! d ! z ! t ! 13 ! 12 ! 11 ! 10

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 768x560 Pixel.
1: Gedehntes Bildformat 768x840 Pixel: Jede zweite Bildschirmzeile wird zweimal gedruckt.

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

13-10: Länge des folgenden IS (0-15 Bytes).

Die drei vordefinierten IS mit Steuerbyte sehen so aus:

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

Achtung: Vor Ausführung der Hardcopy mit Siemens PT-88 muß der

Drucker mit <ESC> 'A2w' auf 12 CPI und mit <ESC> 'A08x' auf 8/72 Zoll Zeilenvorschub geschaltet werden.

IV.6 Spooler und zweite Bildschirmseite

Im Non-Interlaced-Modus wird nur ein 32-KByte-Bereich des Bildschirmspeichers zur Bilddarstellung benötigt. Die zweite Hälfte des 64-KB-Video-RAM's ist frei. Sie kann wahlweise als Spooler, zweite Bildschirmseite oder RAM-Floppy benutzt werden. Die Funktion wird durch den folgenden Befehl umgeschaltet:

ESC ESC '5' a Spooler und Bildschirmseite umschalten.

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 (def).
'4': Schreiben in Hintergrundseite, Spooler aus.
'5': Vorder- und Hintergrundseite vertauschen.

Im Interlaced-Modus hat dieser Befehl keine Wirkung.

Der Host kann die "Hintergrundseite" unsichtbar beschreiben, während auf dem Bildschirm noch die "alte" Vordergrundseite sichtbar ist. Diese Betriebsart wird durch <ESC> <ESC> '54' eingestellt. Nach <ESC> <ESC> '55' werden die Seiten vertauscht, so daß die bisherige Hintergrundseite sichtbar wird und die Vordergrundseite zur Hintergrundseite wird.

Durch Schreiben in die Hintergrundseite und anschließendes Umschalten lassen sich abrupte Bildwechsel durchführen, ohne daß man den Bildaufbau sieht; dies ist z.b. für Grafikbetrieb und Computeranimation sinnvoll (siehe GRIP-3D-Animationssoftware von CONITEC).

Beide Seiten sind unabhängig; das Scrollen erfolgt getrennt, und die Cursorpositionen bleiben auch beim Umschalten erhalten.

Der 32-KByte-**Spooler** dient der Zwischenspeicherung der Daten vom Host zum Drucker. Er faßt etwa 10-15 Textseiten und erlaubt es, während des Ausdruckens gleichzeitig am Rechner weiterzuarbeiten. Das Drucken kann durch folgenden Befehl vorzeitig abgebrochen werden:

ESC ESC 'P' Spooler löschen.

IV.7 Der Lichtgriffel

Der Lichtgriffel-Detektor reagiert auf LOW-Impulse am /LPEN-Eingang; die Empfindlichkeit kann über das Trimmpotentiometer TR1 eingestellt werden. Außerdem läßt sich über den SENSE-Eingang z.b. die Stellung eines Tasters zum Auslösezeitpunkt feststellen.

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

```
    x = ! 1 ! 0 ! nlp ! er ! sen ! la2-0 !
    a = Block-Nummer vertikal + A0h
    b = Block-Nummer horizontal + A0h

  nlp = 0: Lichtgriffel ausgelöst
   er:  Centronics-ERROR-Eingang
  sen:  SENSE-Eingang

 la2-0: Lichtgriffel-Adressbits innerhalb
        eines 8x8-Pixelblocks   -->
```

000	001	
010	011	
100	101	
110	111	

la2-0

Die Lichtgriffelposition ist nur dann gültig, wenn das nlp-Bit = 0 ist. 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!

IV.8 Die Uhr

Die Echtzeituhr läuft in der oberen Statuszeile mit und läßt sich jederzeit stellen und wieder abfragen. Sie verliert jedoch ihre Information, wenn die Spannung abgeschaltet wird. Bei Betrieb an einer PROF-80/180X-CPU-Karte wird die Uhr beim Einschalten automatisch von der Akku-Uhr des PROF aus gestellt. Der Zugriff erfolgt über folgende Befehle:

ESC ESC 'U' a b c Uhr stellen und starten.
ESC ESC 'u' /a b c Uhr abfragen, Parameter wie beim Stellen, jedoch 8. Bit von a,b,c gesetzt!

a = Stunde (BCD-Format) + 20h
b = Minute (BCD-Format) + 20h
c = Sekunde (BCD-Format) + 20h

Beispiel: <ESC> <ESC> 'UC7P' (1Bh 1Bh 55h 43h 37h 50h) stellt 23:17:30 Uhr ein. Die Uhr beginnt zu laufen, sobald das letzte Zeichen ('P') übertragen wurde. Das BCD-Format ist das gleiche, das von CP/M 3.0 für die BDOS-TIME-Funktion verwendet wird.

IV.9 Bildschirmformate

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

ESC ESC '6' a Zeilenabstand einstellen.

a = '8' : Zeilenabstand 8 Pixel (8x8-Zeichenmatrix, default).
'9' : Zeilenabstand 9 Pixel (8x9-Zeichenmatrix).
':' : Zeilenabstand 10 Pixel (8x10-Zeichenmatrix).

Mit diesem Befehl, der im Interlaced-Modus ignoriert wird, können die Textzeilen zur besseren Lesbarkeit auseinandergezogen werden; der Zwischenraum bleibt dunkel. Im Grafikbetrieb (Vektorzeichnen) muß der Zeilenabstand auf 8 Pixel eingestellt sein.

Beispiel: <ESC> <ESC> '69' (1Bh 1Bh 36h 39h) stellt einen Zwischenraum von einer Pixelzeile ein.

ESC ESC '8' a b Textformat wählen.

a = (Zeilenzahl-1) + 20h (def: 37h = 24 Zeilen)
b = (Spaltenzahl-1) + 20h (def: 6Fh = 80 Spalten)

Hiermit wird die Zahl der Zeilen und Spalten auf dem Bildschirm eingestellt. Bei Breitschrift (s.u.) ist die Spaltenzahl halbiert. Bei Formaten, die von dem TVI-Standardformat (80x24) abweichen, müssen einige Softwarepakete angepaßt werden, was aber z.b. bei WORDSTAR kein Problem ist (Zeilen- und Spaltenzahl patchen!).

Wenn mit <ESC> <ESC> '6' ein anderer Zeilenabstand eingestellt wurde, ändern sich die Maximalwerte für Zeilen- und Spaltenzahl dementsprechend.

Beispiel: <ESC> <ESC> '8B ' (1Bh 1Bh 38h 42h 7Fh) wählt das maximale Format von 96x35 (dazu müssen beide Statuszeilen abgeschaltet sein).

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

ESC ESC 'r' a Reset-Befehl mit Wahl der Betriebsart.

a = '0': Non-Interlaced-Modus (default).
a = '1': Interlaced-Modus.

Im Interlaced-Modus sind Spooler und zweite Bildschirmseite abgeschaltet, ein nachleuchtender Monitor (P39-Phosphor) sollte verwendet werden. Die vertikale Auflösung verdoppelt sich auf 560 Bildpunkte.

Achtung: Die Reset-Befehle löschen auch den Host-Puffer! Darum können bis zur Ausführung des Befehls keine weiteren Daten empfangen werden. Es empfiehlt sich, nach dem Senden dieses Befehls eine kurze Verzögerungsschleife in die Übertragung einzubauen.

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

x = ! 0 ! i ! w1 ! w0 ! h1 ! h0 ! v1 ! v0 !

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

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

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

v1-0 = 00: Bild nach oben
01: Bildlage normal (def)
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. 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 beinhaltet Einstellmarken und die aktuellen Zeichensätze. Es erleichtert die optimale Anpassung des Monitors mit dem Befehl <ESC> <ESC> '9'.

------- ENDE VON KAPITEL IV -------

GRIP-4 I N D E X

A1 Adressbelegung

Der Speicherbereich der internen Z80-CPU ist folgendermaßen aufgeteilt:

0000h-3FFFh EPROM (2x16 KByte)
4000h-6000h CPU-RAM (statisch, 8 KByte)
8000h-FFFFh VIDEO-RAM (dynamisch, 2x32 KByte)

Das Video-RAM besteht aus 2 32-KByte-Blöcken (Pages), die über die Adressen 8000h-FFFFh angesprochen und mit dem PAGE-Flag umgeschaltet werden. Auf dem Bildschirm sind (von oben nach unten) zuerst Page 0 und dann Page 1 abgebildet. Im Non-Interlaced-Modus ist allerdings nur ein Teil einer Page sichtbar.

Die Ports auf der Karte belegen die folgenden I/O-Adressen:

Port	R/W	Bit	Name	Funktion
00h	--	--	/STB	Strobe-Signal für die Centronics-Schnittstelle. Ein Ansprechen dieses Ports durch einen Schreib- oder Lesebefehl löst einen LOW-Impuls von ca. 700 ns Dauer auf der /STB-Leitung aus.
10h	W	7	CCON	Ausgangs-Steuerleitung (CCON), auch zum Abschalten der Centronics-Schnittstelle (-> J2) und zum Umschalten der beiden EPROM-Pages. 0: EPROM Page 0 selektiert 1: EPROM Page 1 selektiert
11h	W	7	VOL0	Regelt in Kombination mit VOL1 die Lautstärke des Soundgenerators nach folgender Tabelle: VOL1-0: 00 01 10 11 Ton : aus leise normal laut
12h	W	7	/RTS	Handshake-Signal zur V24-Schnittstelle 0: Bereit zum Datenempfang 1: Warten
13h	W	7	PAGE	Schaltet den Zugriff der CPU zwischen den beiden 32K-Pages des Video-RAM's um. 0: Zugriff auf Video-RAM Page 0. 1: Zugriff auf Video-RAM Page 1.

14h	W	7	/STROBE	Ausgangs-Steuerleitung für langen Strobe-Impuls der Centronics-Schittstelle.
15h	W	7	/SI	Steuersignal für Interlaced-Modus. 0: Interlaced-Modus 1: Non-Interlaced-Modus
16h	W	7	DP	Schaltet im Non-Interlaced-Modus die sichtbare Video-RAM-Seite um. 0: Video-RAM Page 0 sichtbar 1: Video-RAM Page 1 sichtbar
17h	W	7	VOL1	Soundgenerator-Lautstärke (s. VOL0)
20h	R/W	0-7	IDR	Zugriff auf die indirekten STI-Register
21h	R	0	/CTS	Handshake-Signal von der V24/RS232-Schnittstell 0: Senden möglich 1: Warten
21h	R	1	VSYN	50-Hz-Signal für aktuellen Bildzustand 0: Bildfeld sichtbar 1: Bildrand erreicht (Vsync).
21h	R	2	CU/A16	Zusatz-Adressleitung für 128-KB-Bildspeicher (-> J 8).
21h	R	3	BUSY	Handshake-Signal von der Centronics-Schnittstelle: 0: Fertig 1: Warten
21h	R	4	IB	Interrupt von 8255-Port B (Parallel-Tastatur) 0: Kein Interrupt 1: Interrupt von Tastatur ausgelöst
21h	R	5	/SKB	Serieller Tastatureingang (invertiert)
21h	R	6	KDATA	Dateneingang für IBM-Tastatur
21h	R	7	IA	Interrupt von 8255-Port A (ECB-Bus) 0: Kein Interrupt 1: Interrupt von Host-CPU ausgelöst
22h	R/W	0-7	IPRB	Interrupt-Zustandsregister B
23h	R/W	0-7	IPRA	Interrupt-Zustandsregister A
24h	R/W	0-7	ISRB	Interrupt-Serviceregister B
25h	R/W	0-7	ISRA	Interrupt-Serviceregister A

26h	R/W	0-7	IMRB	Interrupt-Maskenregister B
27h	R/W	0-7	IMRA	Interrupt-Maskenregiste A
28h	R/W	0-7	PVR	Zeiger/Vektor-Register
29h	R/W	0-7	TABC	Betriebsmodus für Timer A und B
2Ah	R/W	0-7	TBDR	Timer B Daten (Soundgenerator-Tonhöhe)
2Bh	R/W	0-7	TADR	Timer A Daten
2Ch	R/W	0-7	UCR	Betriebs-Modus der RS232-Schnittstelle
2Dh	R/W	0-7	RSR	V24/RS232-Empfängerstatus
2Eh	R/W	0-7	TSR	V24/RS232-Senderstatus
2Fh	R/W	0-7	UDR	V24/RS232-Daten
30h	--	--	LRS	Ansprechen dieses Ports setzt das LPS-Flag zurück und macht den Lichtgriffel-Detektor wieder "scharf".
40h	R	0-2	LPA0-2	Lichtgriffel-Adressbits innerhalb des 8x8-Pixel-Blocks. Der Block selbst wird von VDC-Register 16 und 17 bestimmt. Jeweils 8 einzelnen Pixeln des Blocks entspricht eine LPA-Adresse nach folgendem Schema:

```
 Spalte:   0 1 2 3 4 5 6 7
           .................
 Zeile: 0 :  000  :  001  :   LPA2-1-0
        1 :.......:.......:
        2 :  010  :  011  :
        3 :.......:.......:
        4 :  100  :  101  :
        5 :.......:.......:
        6 :  110  :  111  :
        7 :.......:.......:
```

40h	R	3	SENSE	Eingangs-Signalleitung für Zusatzfunktionen
40h	R	4-5	JS0-1	Status-Jumperfeld. Jede der beiden Leitungen kann 5 Zustände annehmen: - immer auf 0 (kein Jumper) - immer auf 1 (Stellung 5-6 bzw. 11-12) - wie VOL0 (Stellung 1-2 bzw. 7-8) - wie VOL1 (Stellung 4-5 bzw. 10-11) - wie PAGE (Stellung 2-3 bzw. 8-9)

40h	R	6	/ERROR	Fehlermeldung von der Centronics-Schnittstelle 0: Fehler aufgetreten 1: Alles O.K.
40h	R	7	LPS	Lichtgriffeldetektor 0: Nicht angesprochen 1: Lichtgriffel erkannt (rücksetzbar mit LRS)
50h	W	0-4	VAD	VDC-Registeradresse
52h	W	0-7	VDW	VDC-Daten (schreiben in Register)
53h	R	0-7	VDR	VDC-Daten (lesen aus Register)
60h	W	0-7	DATA	Datenport der Centronix-Schnittstelle
70h	R/W	0-7	PA	8255-Port A (ECB-Bus)
71h	R/W	0-7	PB	8255-Port B (Parallel-Tastatur)
72h	R/W	0-7	PC	8255-Port C (Status für A und B)
73h	W	0-7	PM	Betriebsmodus für Port A,B,C

A2 Jumper

Bei der Auslieferung der Platine sind einige Jumper durch Leiterbahnen in der mit "default" gekennzeichneten Position vor-eingestellt. Bei einer Änderung müssen die entsprechenden Bahnen mit einem scharfen Messer durchtrennt werden.

1. EPROM und RAM

2732 (4 KByte):	J1: 1-2	J11: egal
2764 (8 KByte):	J1: 2-3 (default)	J11: 1-2
27128 (16 KByte):	J1: 2-3 (default)	J11: 1-2
27256 (32 KByte):	J1: 2-3 (default)	J11: 2-3

6116 (2 KByte):	J12: 2-3
6264 (8 KByte):	J12: 1-2

2. Datenausgänge der Centronics-Schnittstelle

Bei speziellen Anwendungen kann es sinnvoll sein, die Ausgänge DATA1-8 in den hochohmigen Zustand zu versetzen. Dafür ist J2 vorgesehen. Es lassen sich die folgenden Modi einstellen:

MODUS 1: Die Ausgänge DATA1-8 sind immer durchgeschaltet; CCON ist eine Ausgangsleitung, die von Flag CCON gesteuert wird. Dieser Modus ist voreingestellt.

MODUS 2: CCON ist ein Eingang, der die Ausgänge DATA1-8 abschaltet, wenn ein HIGH-Pegel angelegt wird.

MODUS 1: J2: Pos. 1-2,3-4 (default)
MODUS 2: J2: Pos. 2-3

3. Jumperfeld

Das Feld J3 besteht aus zwei Jumpern mit je 5 Stellungen, die von der CPU auf den Eingangsleitungen JS0 und JS1 abgefragt werden können. Bedeutung s. Kapitel III.

4. GRIP-COLOR

Bei Verwendung der Farb-Zusatzkarte GRIP-COLOR ist J4 zu öffnen (default = geschlossen)

5. Power-on-Reset

Normalerweise erzeugt GRIP beim Einschalten selbst ein Reset-Signal zur Initialisierung. Dieses Signal liegt an der PCL-Leitung des ECB-Bus (c26) an. Soll das Reset-Signal ausschließlich vom Bus kommen, ist J5 zu öffnen (default = geschlossen); damit ist der interne Power-on-Reset abgeschaltet.

6. V24/RS232-Baudrate

Die Baudrate der seriellen Schnittstelle wird normalerweise intern (von den STI-Timern C und D) erzeugt und steht an der Leitung BAUD zur Verfügung. Alternativ kann die Schnittstelle auch mit externer Baudrate über diese Leitung betrieben werden. Die Konfigurationen werden mit J6 eingestellt.

Baudrate Sender	Empfänger	BAUD-Leitung	J6
Timer C/16	Timer D/16	Timer D	Pos. 1-2,3-4 (default
Timer D/16	Timer D/16	Timer D	Pos. 2-4,3-4
Timer C/16	BAUD/16	Eingang	Pos. 1-2
BAUD/16	BAUD/16	Eingang	Pos. 2-4

7. ECB-Bus-Adresse

Durch Umstecken von J7 lassen sich die externen I/O-Adressen des Status- und Datenports von C0h-C1h auf A0h-A1h ändern. Dies ist erforderlich, wenn C0h oder C1h auf dem System schon anderweitig belegt sind.

I/O-Adressen C0h, C1h: J7: Pos. 1-3,2-4 (default)
I/O-Adressen A0h, A1h: J7: Pos. 1-2,3-4

8. 128-KB-Bildspeicher/ Cursor-Signal

J8 offen: I2 ist A8-Ausgang für RAM's
J8 gesteckt: I2 ist Cursor-Eingang

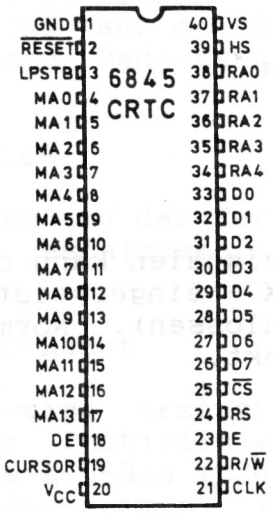
9. Pixel-Taktfrequenz

Zur Synchronisierung mit fremden Videosignalen kann der Bildpunkttakt extern - über die Leitung CKPIX - eingespeist werden. Dazu ist J10 zu öffnen (default = geschlossen). Normalerweise liegt auf CKPIX der interne 16MHz-Systemtakt.

A3 Programmierung des VDC

Der auf GRIP eingesetzte Video-Controller (VDC) 6845 stellt die Verbindung zwischen Mikroprozessorsystem und Bildschirm her. Seine Aufgabe ist es, aus dem Inhalt eines RAM's, auf das auch der Prozessor zugreifen kann, ein Videosignal zu produzieren. Über seine Adressleitungen wird das RAM ständig angesprochen, um den Inhalt auszulesen und damit entweder direkt den Bildschirm oder einen Zeichengenerator anzusteuern. GRIP macht von der ersten Möglichkeit Gebrauch. Zusätzlich liefert der Baustein Signale zur Sychronisierung von Zeilen- und Bildfrequenz, zur Dunkelschaltung und zur Cursorerzeugung. Ein Eingang ist für den Lichtgriffel vorgesehen.

Die Abbildung zeigt die Anschlußbelegung des Bausteins. Die einzelnen Signale haben folgende Funktion:



GND:	Masse
VCC:	+5V-Spannung
D0-D7:	Datenleitungen
E:	Chip-Select
/CS:	Chip-Select
R//W:	Lesen/Schreiben
RS:	Registerauswahl
/RESET:	Rücksetz-Eingang
MA0-13:	Block-Adressen
RA0-4:	Byte-Adressen
CLK:	Takteingang
LPSTB:	Lichtgriffel
DE:	Display hell
CURSOR:	Cursorausgang
VS:	Vertikaler Sync
HS:	Horizontaler Sync

Alle Funktionen lassen sich über die 18 8-Bit-Register des 6845 (R0-R17) steuern. Sie belegen auf GRIP jeweils eine I/O-Adresse zum Einschreiben (52h) und eine zum Auslesen (53h). Über ein internes "Zeigerregister" auf 50h kann das gewünschte Register ausgewählt werden. Zuerst muß dazu die Registernummer (5 Bit) in das Zeigerregister eingeschrieben werden, dann kann man über 52h oder 53h auf das betreffende Register zugreifen.

Die Funktionen von R0-R17 sind im folgenden beschreiben; Rx ist

dabei der in Register x (0..17) einprogrammierte Wert. Die angegebenen Zeiten beziehen sich auf eine 2MHz-Frequenz am CLK-Eingang des 6845; alle internen Frequenzen sind davon abgeleitet.

R0: Zeilenlänge. Die Dauer einer Zeile inklusive Rücklaufperiode beträgt (R0+1)/2 Mikrosekunden. Aus der Zeilendauer ergibt sich dabei die Zeilenfrequenz (in Deutschland 50 Hz * 312.5 Zeilen = 15625 Hz).

R1: Horizontale Auflösung. Die Anzahl der sichtbaren waagerechten Bildpunkte beträgt 8*R1.

R2: Zeilenlage. R2 bestimmt die Position des horizontalen Sync-Impulses in Einheiten von 500 ns, bezogen auf den Zeilenbeginn. Über dieses Register kann die waagerechte Bildposition korrigiert werden: ein höherer Wert schiebt das Bild nach links, ein niedrigerer nach rechts.

R3: Sync-Breite. Die unteren 4 Bit von (R3) bestimmen die Dauer horizontalen Sync-Impulses in Einheiten von 500 ns. Bei einer eingeschreibenen "0" wird das horizontales Sync-Signal ganz unterdrückt. Die Dauer des vertikalen Sync's ist fest auf 8*(R0+1) Mikrosekunden eingestellt.

Die Summe der Registerinhalte R2+R3 muß immer kleiner sein als R0, sonst gibt es Chaos auf dem Bildschirm.

R4, R5 und **R9**: Zeilenzahl. Die Gesamtzahl der Zeilen auf dem Bildschirm ergibt sich aus (R4+1)*(R9+1)+R5. In R4 sind nur die unteren 7 Bit, in R5 und R9 die unteren 5 Bit signifikant. R9+1 bestimmt dabei die Länge eines "Blocks" d.h. die Anzahl der aufeinanderfolgenden Bytes im Video-RAM, die auf dem Bildschirm untereinander dargestellt werden. Über R4 und R5 wird mit der Zeilenzahl auch gleichzeitig die Bildwiederholrate eingestellt. 50 Hz entsprechen 312 Zeilen, 60 Hz 262 Zeilen pro Halbbild. Mit R5 erfolgt der "Feinabgleich" der Wiederholrate auf die Netzfrequenz, der wegen der sonst auftretenden Interferenzerscheinungen ("Bauchtänze") für ein ruhig stehendes Bild erforderlich ist.

R6: Vertikale Auflösung. R6*(R9+1) bestimmt die Anzahl der sichtbaren senkrechten Bildpunkte. R6 muß kleiner als R4 sein.

R7: Bildlage. Die unteren 7 Bit von R7 legen die Position des vertikalen Sync-Impulses in Einheiten von (R0+1)*(R9+1)/2 Mikrosekunden fest. Mit diesem Register läßt sich das Bild senkrecht verschieben. Ein höherer Wert schiebt es nach oben, ein niedrigerer nach unten. R7 darf nicht größer als R4 sein.

R8: Zeilensprung (Interlaced-Modus). Es gibt drei Modi, die mit den beiden unteren Bits von R8 eingestellt werden können:

00b oder 10b schalten den Zeilensprung ab; dies ist der Normalmodus (Non-Interlaced).

01b erzeugt einen Pseudo-Zeilensprung, bei dem beide Halbbilder identisch sind. Dabei verbessert sich die Bildqualität, das Bild beginnt jedoch leicht zu flimmern (-> P39-Monitor erforderlich). Dieser Modus funktioniert nur, wenn in R0 ein ungerader Wert steht.

Mit 11b wird der echte Zeilensprung-Modus eingeschaltet; dabei ist die Auflösung in vertikaler Richtung verdoppelt. Weil dabei die effektive Bildwechselfrequenz auf die Hälfte reduziert wird, flimmert das Bild mit 25 Hz, so daß die Verwendung eines nachleuchtenden Monitors erforderlich ist. Der Zeilensprung-Modus erfordert einen ungeraden Wert für R0 und R9, einen geraden Wert für R6. Außerdem müssen Bit 0 von R10 und R11 übereinstimmen.

R10, R11: Cursor-Steuerung. Ein Cursor-Signal wird auf einer Anzahl aufeinanderfolgender Zeilen eines Byte-Blocks der mit R9 definierten Länge erzeugt. Bit 0-4 von R10 und R11 bestimmen dabei die Anfangs- und die Endzeile innerhalb des Blocks. Über Bit 5 und 6 von R10 läßt sich der Cursor-Modus einstellen: 00b schaltet den Cursor ein, 01b unterdrückt ihn, 10b und 11b lassen ihn mit unterschiedlicher Geschwindigkeit blinken.

R14, R15: Cursor-Position. Die Nummer des Cursor-Blocks ergibt sich aus R15+256*R14. Multipliziert mit der Blocklänge R9+1 erhält man die Adresse des ersten Bytes von diesem Block.

R12, R13: RAM-Startadresse. Der Block ganz oben links auf dem Bildschirm hat die Nummer R13+256*R12. Durch Ändern der Startadresse kann das Bild im Speicher verschoben (gescrollt) werden.

R16, R17: Lichtgriffel-Adresse. Der Lichtgriffel wurde zuletzt auf der Bildschirmposition mit der Blocknummer R17+256*R16 durch eine High-Flanke am LPSTB-Eingang ausgelöst.

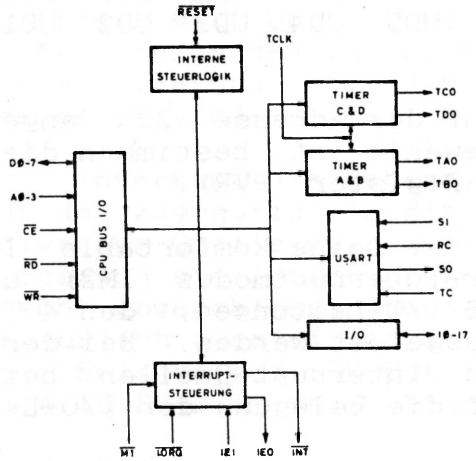
In alle Register, bis auf R16 und R17, lassen sich Daten einschreiben; gelesen werden kann jedoch nur aus den Registern R12-R17.

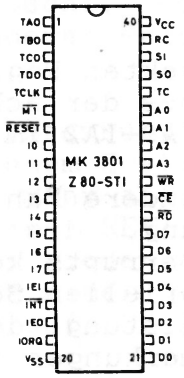
Lit.: Motorola 6845 CRTC Data Manual

A4 Programmierung des STI

STI steht für "Serielle Schnittstelle, Timer, Interrupts" und beschreibt die Funktionen des auf der Graphik-Karte GRIP eingesetzten vielseitigen I/O-Chip's. Die serielle Schnittstelle (USART) kann in mehreren Betriebsarten gefahren werden. Zwei der vier Timer sind einfache Zeitgeber, die beiden anderen lassen sich zusätzlich als Ereigniszähler, als Frequenzteiler oder zur Impulsmessung programmieren. Schließlich enthält der STI noch acht Ein/Ausgänge, die Interrupts auslösen können.

Die folgenden Abbildungen zeigen die innere Organisation und die Anschlußbelegung.





Der Baustein belegt 16 Portadressen; auf GRIP sind dies die Adressen 20h-2Fh. Seine 23 internen Steuerregister sind in 15 direkte (R1-R15) und 8 indirekte (IR0-IR7) aufgeteilt. Die folgende Tabelle zeigt die Belegung der Register.

Adresse	Nummer	Name	Bit 7	6	5	4	3	2	1	0
20h	IR0	SCR	SC7	SC6	SC5	SC4	SC3	SC2	SC1	SC0
20h	IR1	TDDR	TDD7	TDD6	TDD5	TDD4	TDD3	TDD2	TDD1	TDD0
20h	IR2	TCDR	TCD7	TCD6	TCD5	TCD4	TCD3	TCD2	TCD1	TCD0
20h	IR3	AER	AEP7	AEP6	AEP5	AEP4	AEP3	AEP2	AEP1	AEP0
20h	IR4	IERB	IEP5	IEP4	IETC	IETD	IEP3	IEP2	IEP1	IEP0
20h	IR5	IERA	IEP7	IEP6	IETA	IERB	IERE	IEXB	IEXE	IETB
20h	IR6	DDR	DDP7	DDP6	DDP5	DDP4	DDP3	DDP2	DDP1	DDP0
20h	IR7	TCDC	TARS	TCC2	TCC1	TCC0	TBRS	TDC2	TDC1	TDC0

Adresse	Nummer	Name	Bit 7	6	5	4	3	2	1	0
21h	R1	GPIP	DP7	DP6	DP5	DP4	DP3	DP2	DP1	DP0
22h	R2	IPRB	IPP5	IPP4	IPTC	IPTD	IPP3	IPP2	IPP1	IPP0
23h	R3	IPRA	IPP7	IPP6	IPTA	IPRB	IPRE	IPXB	IPXE	IPTB
24h	R4	ISRB	ISP5	ISP4	ISTC	ISTD	ISP3	ISP2	ISP1	ISP0
25h	R5	ISRA	ISP7	ISP6	ISTA	ISRB	ISRE	ISXB	ISXE	ISTB
26h	R6	IMRB	IMP5	IMP4	IMTC	IMTD	IMP3	IMP2	IMP1	IMP0
27h	R7	IMRA	IMP7	IMP6	IMTA	IMRB	IMRE	IMXB	IMXE	IMTB
28h	R8	PVR	IV7	IV6	IV5	VR4	ISE	IRA2	IRA1	IRA0
29h	R9	TABC	TAC3	TAC2	TAC1	TAC0	TBC3	TBC2	TBC1	TBC0
2Ah	R10	TBDR	TBD7	TBD6	TBD5	TBD4	TBD3	TBD2	TBD1	TBD0
2Bh	R11	TADR	TAD7	TAD6	TAD5	TAD4	TAD3	TAD2	TAD1	TAD0
2Ch	R12	UCR	DIV	WL1	WL0	ST1	ST0	PEN	PEO	DMAE
2Dh	R13	RSR	RBF	ROER	PER	FER	SBD	MIP	SSE	RXE
2Eh	R14	TSR	TBE	TUER	AT	EOT	BRK	SO1	SO0	TXE
2Fh	R15	UDR	UD7	UD6	UD5	UD4	UD3	UD2	UD1	UD0

Die indirekten Register werden auf der Adresse 20h angesprochen. Welches der acht gerade ausgewählt ist, bestimmen die drei Adressbits IA0-IA2 im Zeiger/Vektor-Register (PVR).

Eine besondere Fähigkeit der STI ist seine komfortable Interruptbehandlung, die den Z80-Vektor-Interruptmodus (IM2) unterstützt. Interrupts können von den 8 I/O-Leitungen, den 4 Timern oder der seriellen Schnittstelle ausgelöst werden. Bei der folgenden Auflistung der 16 möglichen Interrupt-Quellen beziehen sich die Bemerkungen im Klammern auf die Belegung der I/O-Leitungen bei GRIP.

Priorität	Vektor	Name	Bedeutung
1 (höchste)	1Eh	P7	I/O-Leitung 7 (ECB-Bus)
2	1Ch	P6	I/O-Leitung 6 (Extern)
3	1Ah	TA	Nulldurchgang Timer A
4	18h	RB	Daten empfangen
5	16h	RE	Fehler beim Empfang
6	14h	XB	Daten gesendet
7	12h	XE	Fehler beim Senden
8	10h	TB	Nulldurchgang Timer B
9	0Eh	P5	I/O-Leitung 5 (Tastatur seriell)
10	0Ch	P4	I/O-Leitung 4 (Tastatur parallel)
11	0Ah	TC	Nulldurchgang Timer C
12	08h	TD	Nulldurchgang Timer D
13	06h	P3	I/O-Leitung 3 (BUSY-Leitung)
14	04h	P2	I/O-Leitung 2 (Cursor)
15	01h	P1	I/O-Leitung 1 (Display hell)
16 (letzte)	00h	P0	I/O-Leitung 0 (CTS-Eingang)

Jede der Quellen erzeugt einen eigenen Vektor, dessen untere 5 Bits in der obigen Tabelle angegeben sind. Die oberen 3 Bits (IV5-IV7) sind allen gemeinsam und lassen sich über das PVR-Register programmieren. Die Steuerung der Interrupts erfolgt über 4 Kontroll- und Statusbits (IE, IM, IP, IS). Sie sind über 8 Register für jede Quelle getrennt einstellbar und abfragbar und haben folgende Bedeutung:

IExx: Interrupt einschalten für Quelle xx. Eine "1" ermöglicht den Interrupt, eine "0" schaltet ihn ab.

IMxx: Interrupt maskieren. Ein ausgelöster Interrupt wird nur dann zur CPU geschickt (aktiviert), wenn das zugehörige IM-Bit auf "1" steht.

IPxx: Interrupt ausgelöst. Eine "1" signalisiert den ausgelösten Interrupt. Das Bit wird durch Einschreiben einer "0" oder durch Aktivieren des Interrupts wieder gelöscht. Voraussetzung für die Aktivierung ist, daß die IP-Bits höherer Priorität sämtlich gelöscht sind.

ISxx: Interrupt in Behandlung. Ist ein Interrupt aktiviert, springt das zugehörige IS-Bit auf "1". In diesem Zustand kann von der gleichen Quelle kein weiterer Interrupt mehr aktiviert werden. Sobald die Interrupt-Routine durch eine RETI-Instruktion (EDh 4Dh) abgeschlossen wurde, geht das Bit wieder auf "0"; es kann natürlich auch direkt durch Einschreiben einer "0" gelöscht werden. Solange das ISE-Bit im PVR-Register auf "0" steht, bleiben alle IS-Bits im gelöschten Zustand.

Die 8 Ein/Ausgangsleitungen der STI können auf jeder Flanke eines Eingangssignals einen Interrupt erzeugen. Die aktive Flanke wird dabei durch das zugehörige AEP-Bit im AER-Register bestimmt; bei einer "1" wird der Interrupt auf der steigenden, bei einer "0" auf der fallenden Flanke ausgelöst.

Direkter Lese- oder Schreibzugriff auf diese Leitungen erfolgt über das GPIP-Register. Mit dem DDR-Register können die Datenrichtungen einzeln bestimmt werden. Eine "1" im DDP-Bit macht die betreffende Leitung zum Ausgang, eine "0" zum Eingang.

Die vier Timer A, B, C und D sind Abwärtszähler mit programmierbarer Zeitkonstante. Der Takt wird über die Kontrollflags TAC0-2, TBC0-2, TCC0-2 und TDC0-2 nach folgender Tabelle eingestellt:

Bit	TxC2	TxC1	TxC0	Takt
	0	0	0	Timer gestoppt
	0	0	1	1 MHz
	0	1	0	400 kHz
	0	1	1	250 kHz
	1	0	0	80 kHz
	1	0	1	62.5 kHz
	1	1	0	40 kHz
	1	1	1	20 kHz

Die Werte beziehen sich auf eine Frequenz von 4 MHz am Takteingang der STI. Zusätzlich können die Timer A und B die Pulslänge externer Signale messen, die über die Eingangsleitungen 4 und 3 eingespeist werden. Dazu sind die Bits TAC3 bzw. TBC3 auf "1" zu setzen. Im gestoppten Zustand (TxC0-2 auf "0") liefern die externen Eingänge den Timer-Takt. Diese Features werden auf GRIP nicht benutzt, da die I/O-Leitungen 3 und 4 schon anderweitig belegt sind; TAC3 und TBC3 sollten darum immer auf "0" programmiert werden.

Über die Timer-Datenregister TADR, TBDR, TCDR, TDDR läßt die Zeitkonstante programmieren, die bei jedem Nulldurchgang der Abwärtszähler neu geladen wird. Gleichzeitig wechselt das Signal am jeweiligen Timerausgang. Damit ergibt sich ein Rechtecksignal mit der Frequenz

$$f = T/(2*Z),$$

wobei T der eingestellte Timer-Takt, Z die Zeitkonstante ist. Beim Auslesen der Timer-Datenregister erhält man den augenblicklichen Zählerstand. Die Ausgangsfrequenz von Timer B wird auf GRIP für den Soundgenerator benutzt, Timer C und D liefern die Baudraten für Sender und Emfänger der seriellen Schnittstelle.

Diese Schnittstelle kann synchron oder asynchron betrieben werden. Im Synchronbetrieb wird der Empfänger durch ein vorangestelltes Sync-Byte auf den eintreffenden Datenstrom synchronisiert. Dieses Byte steht im SCR-Register und wird in Übertragungspausen ständig gesendet. Der (üblichere) Asynchronbetrieb erfolgt mit einem Startbit; die Zahl der Stopbits ist variabel. Betriebsarten und Status lassen sich über die Kontrollbits in den Registern UCR, RSR und TSR einstellen bzw. auslesen.

ST1	ST0	Betriebsart
0	0	Synchron
0	1	Asynchron, 1 Stopbit
1	0	Asynchron, 1 1/2 Stopbits
1	1	Asynchron, 2 Stopbits

WL1	WL0	Wortlänge für Sender und Empfänger
0	0	8 Bit
0	1	7 Bit
1	0	6 Bit
1	1	5 Bit

SO1	SO0	Ruhezustand des seriellen Ausgangs
0	0	Hochohmig
0	1	Low ("0")
1	0	High ("1")
1	1	Verbunden mit Empfänger-Eingang

An das gesendete Datenwort kann ein Paritätsbit angehängt werden, um Übertragungsfehler zu erkennen. Dieses Bit ergänzt die Anzahl der "Einsen" im Datenwort auf einen geraden oder ungeraden Wert. Im Asynchronbetrieb ist gerade Parität üblich. Der Paritäts-Modus wird mit PEN und PEO bestimmt.

PEN	PEO	Parität
0	0	Kein Paritätsbit
0	1	Kein Paritätsbit
1	0	Ungerade Parität
1	1	Gerade Parität

Über die restlichen Flags lassen sich bestimmte Zustände der Schnittstelle steuern oder abfragen.

DIV	Teilerfaktor der Takteingänge
0	Faktor 1
1	Faktor 16
DMAE	DMA-Handshake-Betrieb (DMA Enable)
RBF	Zeichen empfangen (Buffer full)
ROER	Empfänger nicht bedient (Overrun Error)
PER	Paritätsfehler
FER	Stopbit nicht erkannt (Frame Error)
SBD	Wort/Unterbrechung erkannt (Found/Search or Break Detect
MIP	Wort gesucht (Match/Character in Progress)
SSE	Sync-Wort ausblenden (Sync Strip Enable)
RXE	Empfänger aktivieren (Receiver Enable)
TBE	Zeichen gesendet (Buffer empty)
TUER	Sender nicht bedient (Underrun Error)

AT	Daten ständig senden (Auto Turnaround)
EOT	Senden beenden (End of Transmission)
BRK	Unterbrechungssignal senden (Break)
TXE	Sender aktivieren (Transmitter enable)

Im DMA-Handshake-Betrieb liegen die Signale RBF und TBE an den I/O-Leitungen 0 und 1. Auf GRIP ist DMA-Betrieb nicht vorgesehen, so daß das DMAE-Bit immer auf "0" programmiert werden muß. RBF springt auf "1", sobald ein empfangenes Datenbyte zum Lesen zur Verfügung steht. Geht TBE auf "1", kann ein Byte zum Senden eingeschrieben werden. Lesen und Schreiben erfolgt über das UDR-Register.

Literatur: Mostek, STI MK3801 Data Manual

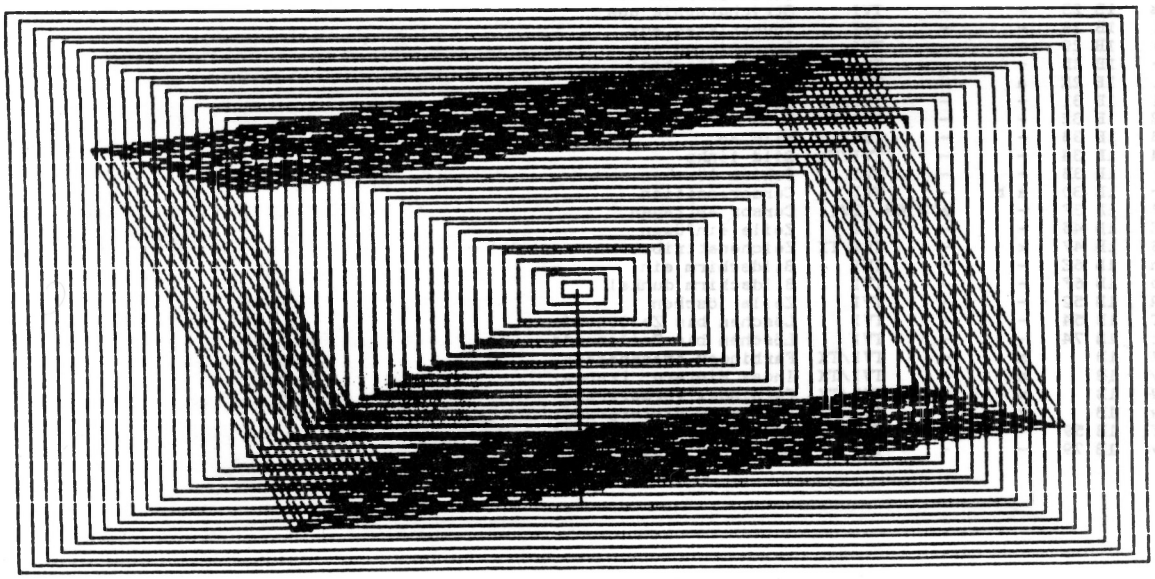
----- K o m m a n d o - Ü b e r s i c h t -----

Bedeutung der Parameter:
- a,b,c = ASCII-Zeichen (20h..7Fh)
- x,y,z = Binärer 8-Bit-Code (00h..FFh)
- aa,bb = 8-Bit-Wort im Hex-Code
- bbbb = 16-Bit-Wort im Hex-Code
- xx = Binäres 16-Bit-Wort
- CR = Carriage Return (0Dh)

Befehl	Hex	Parameter	Antwort	Funktion
ESC ESC 0	1B 1B 30	a	-	V24-Empfängerbaudrate
ESC ESC 1	1B 1B 31	a	-	V24-Senderbaudrate
ESC ESC 2	1B 1B 32	x	-	V24-Kontrollflags
ESC ESC 3	1B 1B 33	a	-	Tastatur-Baudrate
ESC ESC 4	1B 1B 34	a	-	Tastatur-Kanal
ESC ESC 5	1B 1B 35	a	-	Page/Spooler-Zuordnung
ESC ESC 6	1B 1B 36	a	-	Zeilenabstand
ESC ESC 7	1B 1B 37	a	-	Zeichensatz
ESC ESC 8	1B 1B 38	a b	-	Textformat
ESC ESC 9	1B 1B 39	x	-	Monitor-Anpassung
ESC ESC A	1B 1B 41	-	-	TVI-Grundmodus
ESC ESC a	1B 1B 61	-	-	TVI-Farbmodus
ESC ESC C	1B 1B 43	a	-	Zielkanal wählen
ESC ESC c	1B 1B 63	a	-	Quellenkanal wählen
ESC ESC D	1B 1B 44	aa	-	Hex-Byte senden
ESC ESC d	1B 1B 64	x	-	Byte senden
ESC ESC E	1B 1B 45	-	aa	Hex-Byte empfangen
ESC ESC e	1B 1B 65	-	x	Byte empfangen
ESC ESC F	1B 1B 46	aa.. CR	-	Hex-Byte-Folge senden
ESC ESC f	1B 1B 66	xx y z..	-	String senden
ESC ESC G	1B 1B 47	aaaa	bb.. CR	Hex-Bytes empfangen
ESC ESC g	1B 1B 67	xx	y z..	String empfangen
ESC ESC H	1B 1B 48	-	-	Hardcopy an Drucker
ESC ESC h	1B 1B 68	a	-	Hardcopy-Vorwahl
ESC ESC I	1B 1B 49	aaaa	-	Video-RAM-Adresse
ESC ESC i	1B 1B 69	xx	-	Video-RAM-Adresse
ESC ESC J	1B 1B 4A	a	-	Zeichen-Adresse
ESC ESC L	1B 1B 4C	-	x y z	Lichtgriffel-Position
ESC ESC M	1B 1B 4D	aa	bb	Userprogramm starten
ESC ESC m	1B 1B 6D	x	-	Userprogramm starten
ESC ESC O	1B 1B 4F	aa bb	-	Direkte Portausgabe
ESC ESC o	1B 1B 7F	x y	-	Direkte Portausgabe
ESC ESC P	1B 1B 50	-	-	Spooler löschen
ESC ESC p	1B 1B 70	aaaa		Systempatch-Adresse
ESC ESC Q	1B 1B 51	-	-	Monitor-Testbild
ESC ESC R	1B 1B 52	-	-	Karte zurücksetzen
ESC ESC r	1B 1B 52	a	-	Modus umschalten
ESC ESC S	1B 1B 53	-	-	Statuszeile oben
ESC ESC s	1B 1B 73	-	-	Statuszeile unten
ESC ESC T	1B 1B 54	-	-	Tektronix-Grundmodus
ESC ESC U	1B 1B 55	h m s	-	Uhr stellen
ESC ESC u	1B 1B 75	-	h m s	Uhr abfragen
ESC ESC v	1B 1B 76	a	-	Farbebene wählen
ESC ESC W	1B 1B 57	a r g b	-	Farbton für FG=0
ESC ESC w	1B 1B 77	a r g b	-	Farbton für FG=1
ESC ESC X	1B 1B 58	-	-	Tektronix-Farbmodus
ESC ESC Y	1B 1B 57	a r g b	-	Blink-Farbton FG=0
ESC ESC y	1B 1B 77	a r g b	-	Blink-Farbton FG=1
ESC $	1B 24	-	- TVI	Strichgraphik ein
ESC %	1B 25	-	- TVI	Strichgraphik aus
ESC (	1B 28	-	- TVI	Invertieren aus
ESC)	1B 29	-	- TVI	Invertieren ein
ESC +	1B 2B	-	- TVI	Schirm löschen
ESC .	1B 2E	a	- TVI	Cursorattribute
ESC 1	1B 31	-	- TVI/TX	Bild auf
ESC 2	1B 32	-	- TVI/TX	Bild rechts
ESC 3	1B 33	-	- TVI/TX	Bild ab
ESC 4	1B 34	-	- TVI/TX	Bild links
ESC ;	1B 3B	-	- TVI	Schirm löschen
ESC =	1B 3D	a b	- TVI	Cursorposition
ESC ?	1B 3F	-	a b CR TVI	Cursor abfragen
ESC E	1B 45	-	- TVI	Zeile einfügen
ESC G	1B 47	x	- TVI/TX	Zeichenattribute
ESC n	1B 6E	-	- TVI	Bildschirm ein
ESC o	1B 6F	-	- TVI	Bildschirm dunkel
ESC R	1B 52	-	- TVI	Zeile löschen
ESC T	1B 54	-	- TVI	Löschen bis Zeilenende
ESC t	1B 74	-	- TVI	Löschen bis Zeilenende
ESC V	1B	a	- TVI/TX	Farbindex transparent
ESC v	1B	a	- TVI/TX	Index Hinterg. transp.
ESC W	1B	x	- TVI/TX	Farbindex Vordergrund
ESC w	1B	x	- TVI/TX	Farbindex Hintergrund
ESC Y	1B 59	-	- TVI	Löschen bis Seitenende
ESC y	1B 79	-	- TVI	Löschen bis Seitenende

Befehl	Hex	Parameter	Antwort	Funktion
ESC ^L	1B 0C	-	- TX	Schirm löschen
ESC F	1B 46	-	- TX	Fläche füllen
ESC `	1B 60	-	- TX	Vektor durchgehend
ESC a	1B 61	-	- TX	Vektor gepunktet
ESC b	1B 62	-	- TX	Strichpunkt-Vektor
ESC c	1B 63	-	- TX	Kurzstrich-Vektor
ESC d	1B 64	-	- TX	Langstrich-Vektor
^G	07	-	- TVI	Glocke/Sound
^H	08	-	- TVI	Cursor nach links
^I	09	-	- TVI	Cursor nach rechts
^J	0A	-	- TVI	Zeilenvorschub
^K	0B	-	- TVI	Cursor auf
^L	0C	-	- TVI	Schirm initialisier
^M	0D	-	- TVI	Wagenrücklauf
^V	16	-	- TVI	Cursor ab
^Z	1A	-	- TVI	Schirm löschen
^^	1E	-	- TVI	Cursor Home
^_	1F	-	- TVI	Neue Zeile
^G	07	-	- TX	Vektor zeichnen
^M	0D	-	- TX	Alphamodus ein
^Q	11	-	- TX	Vektor schwarz
^R	12	-	- TX	Vektor invertieren
^S	13	-	- TX	Vektor weiß
^Ö	1C	-	- TX	Punktmodus
^U	1D	-	- TX	Vektormodus
^^	1E	-	- TX	Inkrementalmodus
^_	1F	-	- TX	Alphamodus
	20	-	- TX-I	Transparent
A	41	-	- TX-I	Nach rechts
B	42	-	- TX-I	Nach links
D	44	-	- TX-I	Nach oben
E	45	-	- TX-I	Nach rechts oben
F	46	-	- TX-I	Nach links oben
H	48	-	- TX-I	Nach unten
I	49	-	- TX-I	Nach rechts unten
J	4A	-	- TX-I	Nach links unten
P	50	-	- TX-I	Weiß
Q	51	-	- TX-I	Schwarz
R	52	-	- TX-I	Invertiert

```
10 REM ************************************************************************
20 REM ***                                                                  ***
30 REM ***  GRIP-2 Tektronix-Demo mit Hardcopy fuer Epson FX-80              ***
40 REM ***                                                                  ***
50 REM ************************************************************************
60 REM
70 REM
80 REM
90 REM
100 REM verhindere Auto-Linefeed
110 WIDTH 255
120 REM loesche Bildschirm
130 PRINT CHR$(12);
140 REM schalte in Tektronix-Modus
150 PRINT CHR$(27);CHR$(27);"T";
160 REM
170 REM Hauptprogramm
180 REM
190 XOFF=400:YOFF=285
200 XFAKT=300:YFAKT=180
210 FOR K=0 TO 1.2 STEP .03
220 X1=K:Y1=K:GOSUB 470
230 NEXT K
240 FOR K=1 TO .7 STEP -.02
250 X1=K*.6:Y1=K:GOSUB 470
260 NEXT K
270 REM
280 REM Hardcopy
290 REM
300 REM schalte Spooler ein
310 PRINT CHR$(27);CHR$(27);"5";
320 REM definiere Epson FX-80 als Hardcopy-Drucker
330 PRINT CHR$(27);CHR$(27);"h0";
340 REM Hardcopy
350 PRINT CHR$(27);CHR$(27);"H";
360 REM zurueck in TVI-Modus
370 REM
380 REM Ende
390 REM
400 PRINT CHR$(27);CHR$(27);"A";
410 REM positioniere Cursor
420 PRINT CHR$(27);"=5 ";
430 END
440 REM
450 REM Diese Unterprogramm zeichent ein Rechteck auf den Bildschirm
460 REM
470 PRINT CHR$(&H1D);
480 FOR I=1 TO 5
490 X=X1*XFAKT+XOFF:Y=Y1*YFAKT+YOFF:GOSUB 560
500 Z=X1:X1=-Y1:Y1=Z
510 NEXT I
520 RETURN
530 REM
540 REM Dieses Unterprogramm gibt die X/Y-Koordinaten im Tektronix-Format aus
550 REM
560 Y1%=Y\32
570 Y2%=Y MOD 32
580 X1%=X\32
590 X2%=X MOD 32
600 X1%=X1%+&H20:X2%=X2%+&H40
610 Y1%=Y1%+&H20:Y2%=Y2%+&H60
620 PRINT CHR$(Y1%);CHR$(Y2%);CHR$(X1%);CHR$(X2%);
630 RETURN
```



Screen # 0

```
( Initials grip )(   (tektronics-emulation)    LRA 17.07.85 )
   ( ** definitions are complete in standard fig-FORTH ** )

DECIMAL
760   CONSTANT   XMAX          510   CONSTANT   YMAX
380   CONSTANT   MITTE-X       260   CONSTANT   MITTE-Y   -->

ASSEMBLER ( )( swaps high and low bytes of top stack word )
        ( 'flip' mit einem Z80 Assembler z.B. so : )
CODE )(  HL POP    A, L LD    L, H LD    H, A LD    HPUSH JP
END-CODE                                                 -->

        ( mit dem  8080 Assembler von John J. Cassady )
CODE )(  H POP   L A MOV  H L MOV   A H MOV  NEXT 1 - JMP
END-CODE                                                 -->
        ( 'flip' ist auch einfach in HEX-Code zu definieren )
```

Screen # 1

```
( GRIP-2 TREIBER - MASKEN    ysend              LRA 16.07.85 )
CR ." loading grip-2 graphic driver "
DECIMAL
: BINARY       2 BASE ! ;
BINARY
  100000       CONSTANT   YH          ( Y-HI )
 1100000       CONSTANT   YL          ( Y-LO )
  100000       CONSTANT   XH          ( X-HI )
 1000000       CONSTANT   XL          ( X-LO )
   11111       CONSTANT   XYL-CLEAR   ( X Y LO-byte clear )
DECIMAL
: 3LEFT-SHIFT  2 * 2 * 2 * ;          ( arithmetic left shft)
: YSEND ( n - )                       ( Y-Koor. zu GRIP )
            DUP XYL-CLEAR  AND
            YL OR  SWAP  3LEFT-SHIFT
            )(  YH OR  EMIT EMIT       ;            -->
```

Screen # 2

```
( GRIP-2 TREIBER    xsend setxy xy            LRA 16.07.85 )
: XSEND ( n - )                       ( X-Koor. zu GRIP )
            DUP XYL-CLEAR  AND
            XL OR  SWAP  3LEFT-SHIFT
            )(  XH OR  EMIT EMIT  ;

: SETXY ( xn yn - )                   ( X und Y zu GRIP )
      YSEND  XSEND  ;

: XY    ( xn yn - )                   ( mit error checking )
      2DUP  DUP 0(  SWAP  YMAX )  OR    ( check y )
      SWAP  DUP 0(  SWAP  XMAX )  OR    ( check x )
      OR  IF  DROP DROP  R) DROP        ( error )
          ELSE  SETXY                   ( no error )
          THEN  ;
DECIMAL -->
```

Screen # 3

```
( .umrandung  TVI-TEK-Escape-Sequenzen        LRA 16.07.85 )
: .UMRANDUNG     0    0 XY
                 0 YMAX XY
              XMAX YMAX XY
              XMAX    0 XY
                 0    0 XY   ;
HEX
: ESC          1B EMIT   ;          ( TVI )
: CLS          0C EMIT   ;
: TVI          ESC ESC  41 EMIT  ;
: TEK          ESC ESC  54 EMIT  ;
: PUNKT        1C EMIT   ;          ( TEK )
: VEKTOR       1D EMIT   ;
: DIREKT       07 EMIT   ;
: SCHWARZ      11 EMIT   ;
: WEISS        13 EMIT   ;                  DECIMAL     -->
```

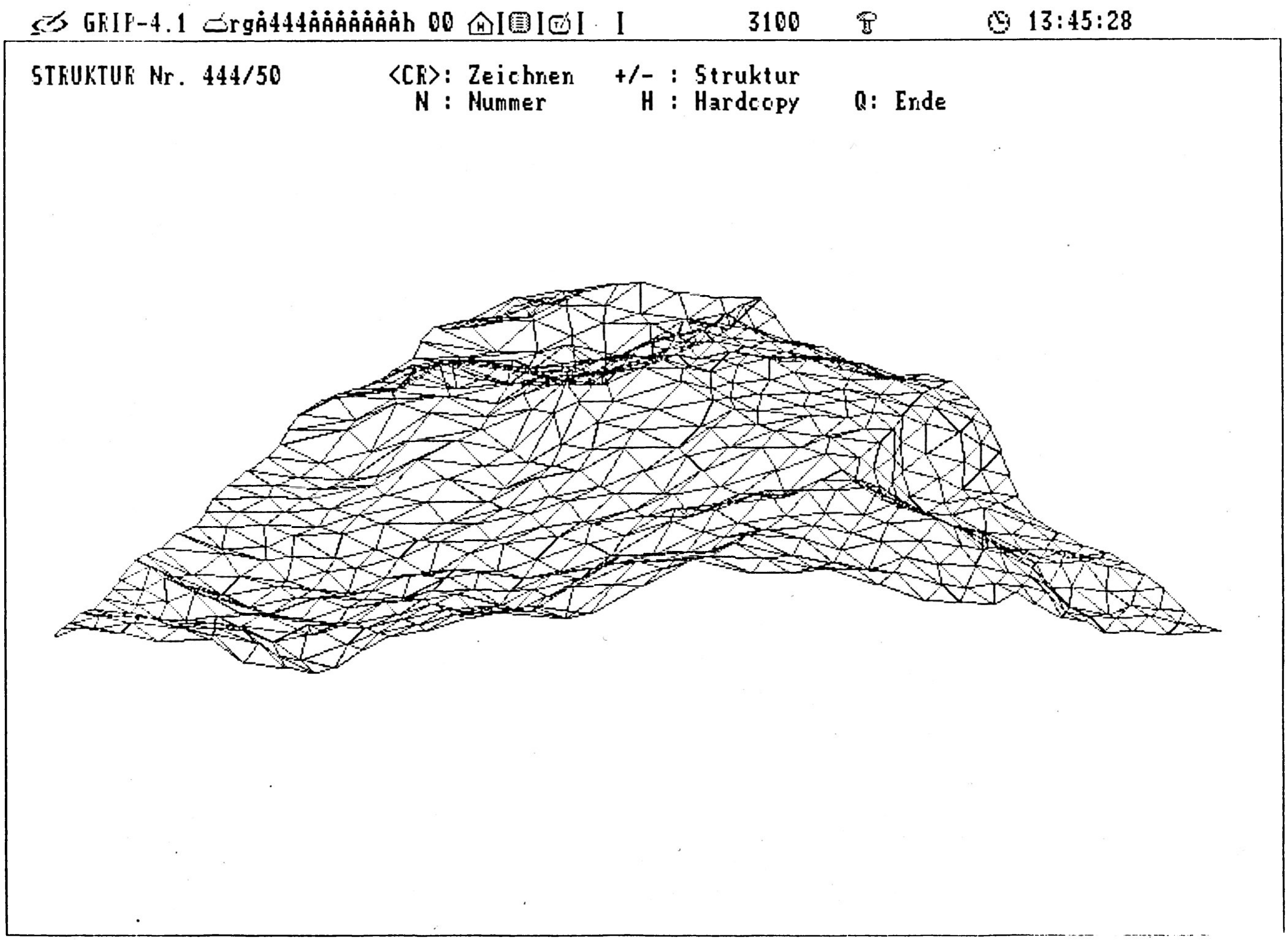
Screen # 4

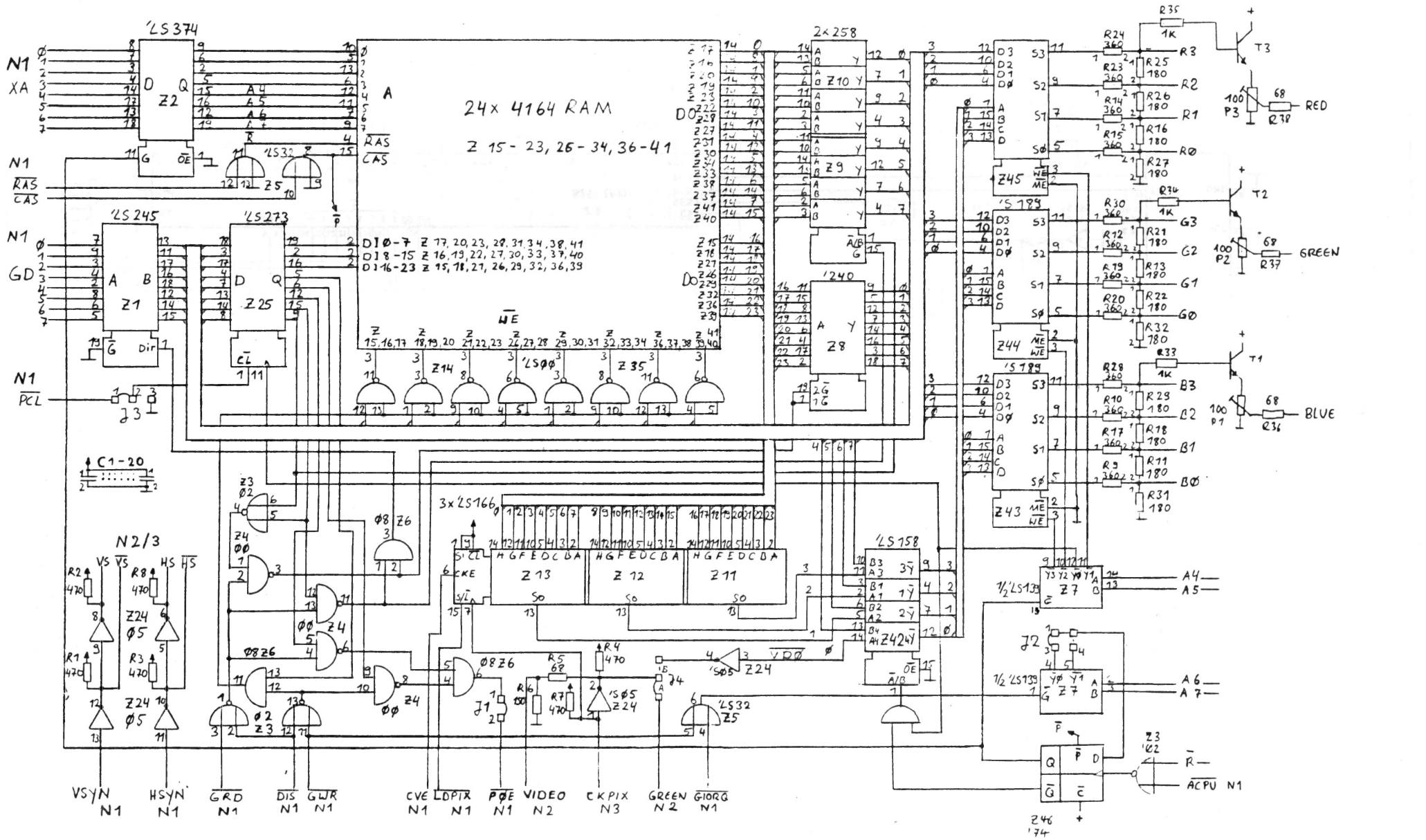
```
( Relative Koordinaten  xco yco .xy home      LRA 14.07.85 )
CR ." loading relative coor. "  ( 'xy' ist die Grundfunktion )
0 VARIABLE   XCO       0 VARIABLE   YCO

: .XY          XCO @  YCO @   XY  ;          ( - )
: HOME         MITTE-X XCO !  MITTE-Y YCO !    ( - )
               SCHWARZ  .XY  WEISS  ;

: MOVETO       ( xn yn - )
               YCO +!  XCO +!  SCHWARZ .XY WEISS  ;
: DRAWTO       ( xn yn - )
               YCO +!  XCO +!  .XY ;

: MM           HOME  ;
: MT           MOVETO  ;
: DT           DRAWTO  ;                  DECIMAL      -->
```

Screen # 5

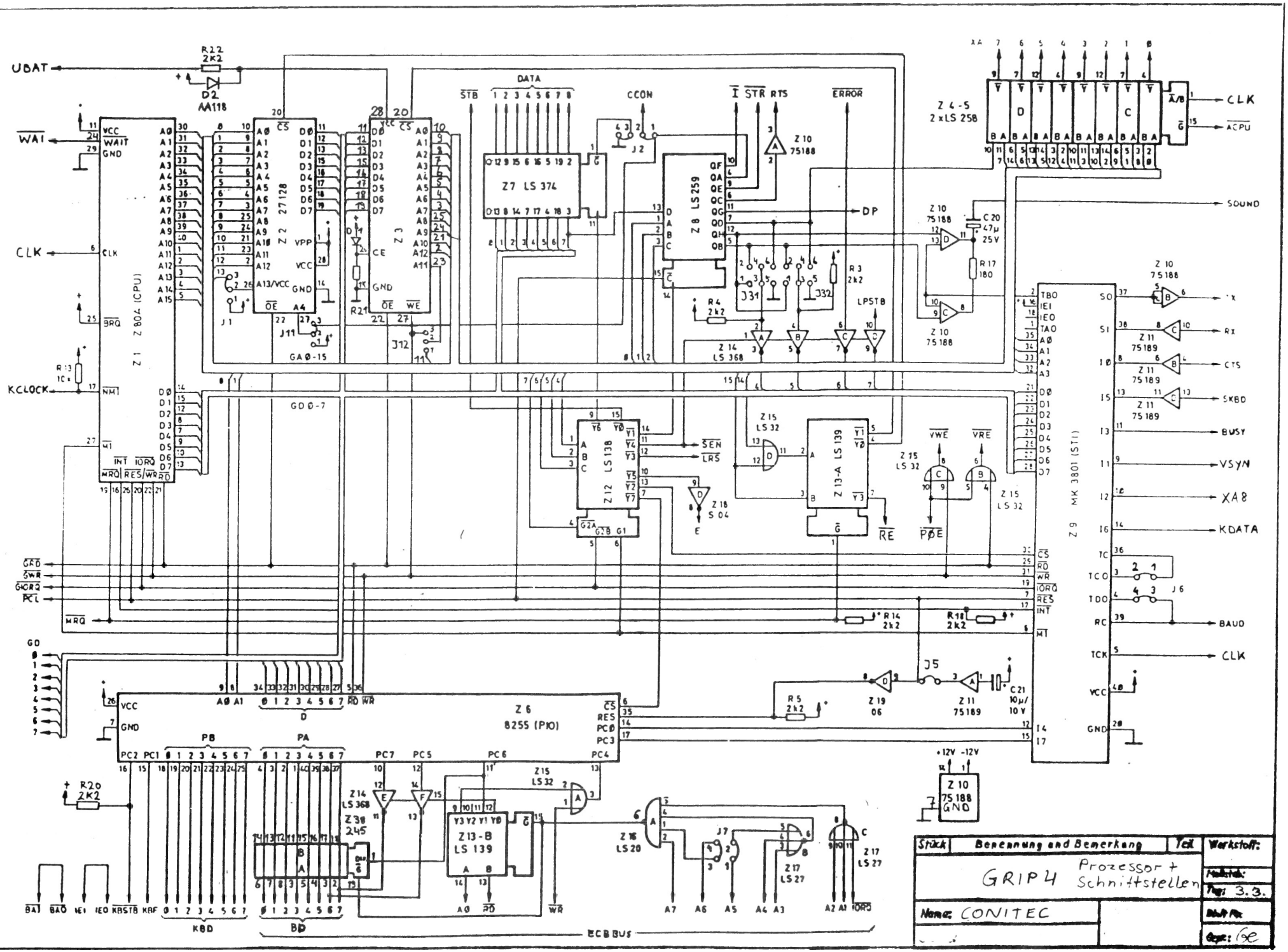
```
( a ap e Demo's relative und abs. Koordinaten   LRA 14.07.85 )
: A     CLS TEK .UMRANDUNG  VEKTOR  DIREKT    ;  ( Anfang )
: AP    CLS TEK .UMRANDUNG  PUNKT             ;  ( Anf. P.Mod )
: E         TVI  ;                                 ( Ende )

( Beispiel relative K.  Quadrat )
: EREL  A  HOME   -40 -40 MT      0  80  DT     80  0  DT
                                  0 -80  DT    -80  0  DT  E ;

( Demo relative K.  'Nadelbaum-Zweig' )
: D1  A  HOME  200 0 DO 150 0 DO  I J MT J I DT  LOOP LOOP E ;

( Demo absolute Koordinaten                      UB  28.06.85 )
: D2 CLS TEK VEKTOR 760 0 DO 380 120  I   0 XY XY  5 +LOOP
                    760 0 DO 380 120  I 240 XY XY  5 +LOOP
                    760 0 DO 380 120  I 480 XY XY  5 +LOOP E ;
```

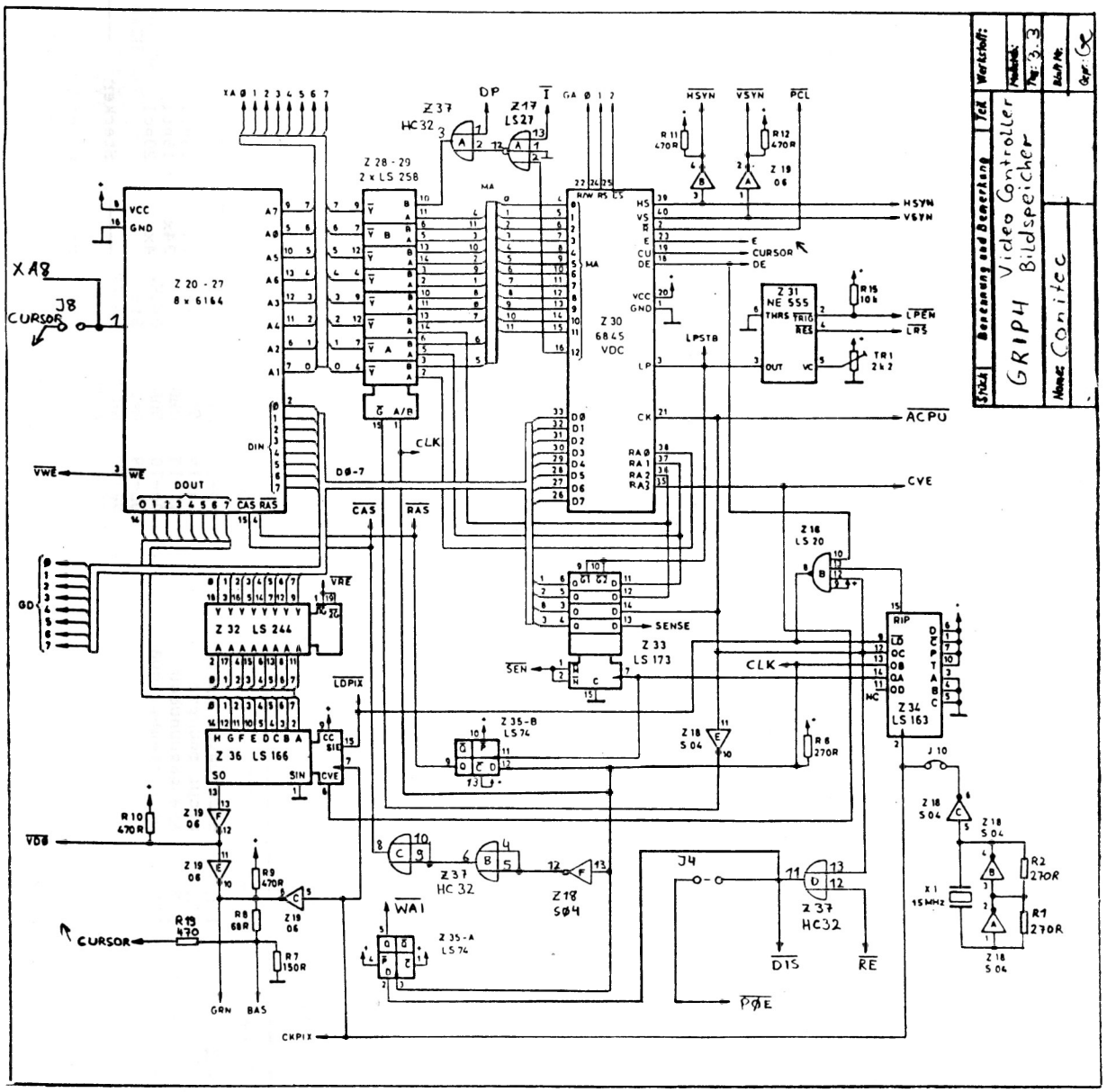
```pascal
program BERG;

type
  Koord = record
            x: 1..767;
            y: 1..539;
          end;
  Triangle = arrayÄ1..3Ü of Koord;
  NewTriangles = arrayÄ1..4Ü of Triangle;

var
  I,J,K,JK: integer;
  FAC1,FAC: integer;
  ALTBERG: arrayÄ1..256Ü of Triangle;
  NEUBERG: arrayÄ1..256Ü of NewTriangles;
  KEY: char;
  NEU,MORE: boolean;


procedure Plot (X,Y: integer);
begin
  write (chr(Y div 32 + 32),chr(Y mod 32 + 96),
         chr(X div 32 + 32),chr(X mod 32 + 64))
end;

procedure Vector (P: Koord);
begin  Plot(P.X,P.Y)  end;

procedure DefDreieck (AX,AY,BX,BY,CX,CY: integer;
                      var T: Triangle);
begin
  TÄ1Ü.X:=AX; TÄ1Ü.Y:=AY;
  TÄ2Ü.X:=BX; TÄ2Ü.Y:=BY;
  TÄ3Ü.X:=CX; TÄ3Ü.Y:=CY;
end;

function Zufall(X: integer): real;
begin
  Zufall:=(Frac((X+FAC1)*9.821+0.211327)-0.5)*FAC/100
end;

procedure Split (A,B: Koord; var C:Koord);
var R1,R2,LENGTH: real;
begin
  R1:=Zufall(A.X+B.X); R2:=Zufall(A.Y+B.Y);
  LENGTH:=Sqrt(Sqr(Int(A.X)-Int(B.X))+
               Sqr(Int(A.Y)-Int(B.Y)));
  C.X:=(A.X + B.X + Trunc(LENGTH*R1)) div 2;
  C.Y:=(A.Y + B.Y + Trunc(LENGTH*R2)) div 2;
  if C.Y<3 then C.Y:=3; if C.Y>530 then C.Y:=530;
end;

procedure NeuDreieck (T: Triangle;var TNEU: NewTriangles);
begin
  TNEUÄ1,1Ü:=TÄ1Ü;
  Split (TÄ1Ü,TÄ2Ü,TNEUÄ1,2Ü);
  Split (TÄ1Ü,TÄ3Ü,TNEUÄ1,3Ü);
  TNEUÄ2,1Ü:=TNEUÄ1,2Ü; TNEUÄ2,2Ü:=TÄ2Ü;
  Split (TÄ2Ü,TÄ3Ü,TNEUÄ2,3Ü);
  TNEUÄ4,1Ü:=TNEUÄ1,3Ü;TNEUÄ4,2Ü:=TNEUÄ2,3Ü;
  TNEUÄ4,3Ü:=TÄ3Ü;TNEUÄ3,1Ü:=TNEUÄ1,3Ü;
  TNEUÄ3,2Ü:=TNEUÄ1,2Ü;TNEUÄ3,3Ü:=TNEUÄ2,3Ü;
end;

procedure ZeichneDreieck (T: Triangle);
begin write (^U);
  Vector(TÄ1Ü);Vector(TÄ2Ü);Vector(TÄ3Ü);Vector(TÄ1Ü);
end;

procedure ZeichneNeuDreieck (TNEU: NewTriangles);
begin
  write(^U); Vector(TNEUÄ1,1Ü);
  Vector(TNEUÄ1,2Ü);Vector(TNEUÄ2,2Ü);Vector(TNEUÄ4,2Ü);
  Vector(TNEUÄ4,3Ü);Vector(TNEUÄ4,1Ü);Vector(TNEUÄ3,2Ü);
  Vector(TNEUÄ3,3Ü);Vector(TNEUÄ3,1Ü);Vector(TNEUÄ1,1Ü);
end;


BEGIN (* Hauptprogramm *)
  write   ('Bitte geben Sie die Nummer der ');
  write   ('Struktur ein (1..999): ');
  read(FAC1); FAC:=50; NEU:=true;

  repeat
    if NEU then begin
      write(^Ä^Ä'A'); ClrScr; MORE:=true; JK:=1;
      GotoXY(3,1); write ('STRUKTUR Nr. ',FAC1,'/',FAC);
      GotoXY(30,1); write ('<CR>: Zeichnen   +/- : Struktur');
      GotoXY(30,2); write ('  N : Nummer       H : Hardcopy');
      write ('    Q: Ende'^Ä^Ä't'^U);
      Plot(0,0);Plot(0,540);Plot(767,540);Plot(767,0);Plot(0,0);
      DefDreieck(30,180,300+trunc(300*ZUFALL(1)),380,
            730,180-trunc(50*ZUFALL(2)),NEUBERGÄ1,1Ü);
      ZeichneDreieck(NEUBERGÄ1,1Ü);
    end;
  read (kbd,KEY);
    case UpCase(KEY) of
      ^M : if MORE and (JK<=64) then begin I:=0;
             for J:=1 to JK do  for K:=1 to 4 do
               if (not NEU) or (K=1) then begin I:=I+1;
                 ALTBERGÄIÜ:=NEUBERGÄJ,KÜ;
               end;
             NEU:=false; JK:=I;
             for J:=1 to JK do begin
               write(^Q); ZeichneDreieck(ALTBERGÄJÜ);
               NeuDreieck(ALTBERGÄJÜ,NEUBERGÄJÜ);
               write(^S); ZeichneNeuDreieck(NEUBERGÄJÜ);
             end;
           end
           else for J:=1 to JK do begin
             write(^Q); ZeichneNeuDreieck(NEUBERGÄJÜ);
             NeuDreieck(ALTBERGÄJÜ,NEUBERGÄJÜ);
             write(^S); ZeichneNeuDreieck(NEUBERGÄJÜ);
             MORE:=true;
           end;

      'H': write (^Ä^Ä'H');   (*HARDCOPY*)
      '+','-': begin
               MORE:=false;
               if KEY='+' then FAC:=FAC+1 else FAC:=FAC-1;
               write(^Ä^Ä'A');
               GotoXY(20,1); write (FAC);
               write (^Ä^Ä't');
             end;
       'N': begin
             write(^Ä^Ä'A');
             GotoXY(16,1); write ('    ');
             GotoXY(16,1); read (FAC1);
             write (^Ä^Ä't'); NEU:=true;
           end;
    end;
  until UpCase(KEY) = 'Q';
  write(^Ä^Ä'A');
END.
```





28.11.84 Color Karte 26.11.84
Gc 21.3.85
1.4.85
17.4.85


UBAT
R22 2K2
D2 MA118
WAI
CLK
KCLOCK
Z1 Z 80A (CPU)
Z2 27128
Z3
GA0-15
GD0-7
DATA
STB
CCON
Z7 LS 374
Z8 LS 259
DP
I STR RTS
ERROR
Z10 75188
Z4-5 2xLS 258
CLK
ACPU
SOUND
R3 2k2
R4 2k2
LPSTB
Z14 LS 368
C20 47µ 25V
R17 180
Z12 LS 138
Z13-A LS 139
Z13-B LS 139
Z15 LS 32
Z18 S 04
SEN
LRS
RE
PDE
VWE
VRE
Z9 MK 3801 (STI)
SO
SI
I0
I5
I3
I1
I2
I6
TC
TCO
TDO
RC
TCK
BUSY
VSYN
XA8
KDATA
BAUD
CLK
J6
R14 2k2
R18 2k2
R5 2k2
R20 2K2
Z19 06
Z11 75189
C21 10µ/ 10 V
J5
Z6 8255 (PIO)
PB
PA
PC7 PC5 PC6 PC4
PC2 PC1 PC0 PC3
Z16 LS 20
Z17 LS 27
J7
+12V -12V
GND
BAI BAO IEI IEO KBSTB KBF
KBD
BD
ECB BUS
Stück
Benennung und Bemerkung
Teil
Werkstoff:
GRIP4 Prozessor + Schnittstellen
Maßstab:
Name: CONITEC
Datum:
Blatt Nr.
Page: 3.3.


GRIPH Video Controller Bildspeicher
Conitec
Z 20 - 27
8 x 6164
Z 28 - 29
2 x LS 258
Z 30
6845
VDC
Z 31
NE 555
Z 32
LS 244
Z 33
LS 173
Z 34
LS 163
Z 36
LS 166
Z 37
HC 32
Z 17
LS 27
Z 18
S 04
Z 19
06
Z 35-A
LS 74
Z 35-B
LS 74
Z 18
LS 20
XA8
J8
CURSOR
VWE
DOUT
DIN
D0-7
CLK
CAS
RAS
VRE
LDPIX
SEN
SENSE
LPSTB
ACPU
CVE
HSYN
VSYN
PCL
DE
LPEN
LRS
WAI
DIS
RE
PDE
CKPIX
GRN
BAS
VDD
R1 270R
R2 270R
R7 150R
R8 68R
R9 470R
R10 470R
R11 470R
R12 470R
R13 470
R15 10k
TR1 2k2
X1 15MHz
J4
J10

S t e c k e r b e l e g u n g

N1: ECB-Bus-Stecker

	a	b	c	Leitung	Typ	Funktion
1:	+5V	+5V	+5V	+5V:	I,U	Betriebsspannung
2:	D5	/POE	D0	+12V,-12V:	I,U	V24-Spannungen
3:	D6	/GRD	D7	-5V:	I,U	nicht benutzt
4:	D3	/GWR	D2	GND:	I,U	gemeinsame Masse
5:	D4	GD0	A0			
6:	A2	GD1	A3	D0-D7:	IO,TS	Externer Datenbus
7:	A4	XA6	A1	A0-A7:	I	Externer Adressbus
8:	A5	XA4	xxx			
9:	A6	GD2	A7	/RD,/WR:	I	Extern Lesen/Schreibe
10:	xxx	GD3	xxx	/IORQ:	I	Externe Ein/Ausgabe
11:	xxx	XA0	IEI			
12:	xxx	XA1	xxx	/PCL:	I/O,OD	Karte rücksetzen
13:	+12V	/DIS	xxx			
14:	-12V !	GD4	D1	IEI,IEO:	I,O	INT-Kette
15:	-5V	GD5	-	/BAI,/BAO:	I,O	DMA-Kette
16:	-	XA5	IEO			
17:	xxx	XA7	xxx	GD0-7:	IO,TS	Graphik-Datenbus
18:	xxx	GD6	xxx	XA0-7:	O,TS	Graphik-Adressbus
19:	-	GD7	xxx			
20:	xxx	/CAS	xxx	/DISP:	O	8000h-FFFFh Select
21:	xxx	XA3	xxx	/POE:	I/O	Grundebene Select
22:	-	XA2	/WR	/GRD,/GWR:	IO,TS	Intern lesen/schreibe
23:	/BAI !	CVE	-	/GIORQ:	IO,TS	Interne Ein/Ausgabe
24:	xxx	/GIORQ	/RD	CVE:	O	Zeilenausblendung
25:	/BAO !	/RAS	xxx	/ACPU:	O	CPU-RAM-Zyklus
26:	xxx	/VD0	/PCL			
27:	/IORQ	VSYN	xxx	/RAS,/CAS:	IO,TS	RAM-Steuersignale
28:	xxx	HSYN	xxx			
29:	xxx	/ACPU	xxx	CKPIX:	I/O	Bildpunkttakt
30:	xxx	/LDPIX	xxx	/LDPIX:	O	Ladetakt
31:	xxx	CKPIX	xxx	/VD0:	O,OD	Video-Signal
32:	GND	GND	GND	VSYN,HSYN:	O	Sync-Signale

xxx = reserviert
\- = zur freien Verwendung
! = abweichend von KONTRON-Busbelegung

Typ:
- I = Eingang
- O = Ausgang
- ST = Schmitt-Trigger
- OD = Offener Kollektor/ Open Drain
- TS = Tri-State
- V24 = V24-Spannungspegel
- I/O = Eingang/Ausgang, wählbar
- IO = Eingang/Ausgang, umschaltbar
- A = Analog-Ein/Ausgang
- U = Betriebsspannung

N2: Video-Signale

			Leitung	Typ	Funktion
01-02	GND	BAS	GND:	O,U	gemeinsame Masse
03-04	VSYN	/VSYN	BAS:	O,OD	Video-Signal
05-06	HSYN	/HSYN	GREEN:	O,OD	Video-Signal TTL
07-08	GND	BAS	VSYN,HSYN:	O	pos. Sync-Signale
09-10	GND	GREEN	/VSYN,/HSYN:	O,OD	neg. Sync-Signale
11-12	GND	--			
13-14	GND	CKPIX	CKPIX:	I/O	Bildpunkttakt (s.J10)
15-16	GND	SOUND	SOUND:	O,A	NF-Ausgang

Für Monitore mit BAS-Eingang müssen die Leitungen BAS, /VSYN und /HSYN (Pins 2,4,6) zur Erzeugung des BAS-Signals miteinander verbunden werden.

N3: Schnittstellen

			Leitung	Typ	Funktion
01-02	+5V	+5V	+5V,-5V,	O,U	Versorgungsspannungen
03-04	DATA1	DATA2	+12V,-12V:	O,U	für externe Geräte
05-06	DATA3	DATA4	GND:	O,U	gemeinsame Masse
07-08	DATA5	DATA6			
09-10	DATA7	DATA8	DATA1-8:	O,TS	Centronix-Daten
11-12	/STB	BUSY	/STB:	O	Übergabe-Impuls, kurz
13-14	/INIT	/ERROR	/STROBE:	O	Übergabe-Impuls, lang
15-16	/STROBE	/CCON	BUSY:	IO,TS	Nicht empfangsbereit
17-18	GND	GND	/INIT:	O,OD	Initialisieren
19-20	GND	SOUND	/ERROR:	I	Fehler
21-22	---	SENSE	/CCON:	I/O	Steuerleitung (s.J2)
23-24	/LPEN	-5V			
25-26	+12V	-12V			
27-28	SKBD	BAUD	TX:	O,V24	V24-Datenausgang
29-30	TX	RX	RX:	I,ST	V24-Dateneingang
31-32	RTS	CTS	RTS:	O,V24	Handshake-Ausgang
33-34	GND	GND	CTS:	I,ST	Handshake-Eingang
			BAUD:	I/O	16facher Baudraten-Takt (s.J6)

N4: Schnittstellen

			KBD1-8:	IO,TS	Tastatur-Daten
			/KBSTB:	IO,TS	Übergabe-Impuls
01-02	+5V	+5V	KBF:	IO,TS	Puffer belegt
03-04	KBD1	KBD2	SKBD:	I,ST	Serieller Tastatur-Eingang
05-06	KBD3	KBD4			
07-08	KBD5	KBD6			
09-10	KBD7	KBD8	KDATA:	I	IBM-Tastatur-Eingang
11-12	/KBSTB	KBF	KCLOCK:	I	IBM-Tastatur-Takt
13-14	KDATA	KCLOCK			
15-16	---	---			
17-18	GND	GND			
19-20	+12V	SOUND			
21-22	2 MHz	SENSE	SOUND:	O,A	NF-Ausgang
23-24	/LPEN	-5V			
25-26	GND	GND	/LPEN:	I,A	Lichtgriffel-Impuls

GRIP-4-COLOR Steckerbelegung

N1: Grafik-Bus (ECB-Bus, b-Leiste, Belegung wie GRIP-4)

N2: Video-Signale

				Leitung	Typ	Funktion
01-02	GND	VIDEO	:	RED:	O,A	Video-Signal rot
03-04	VS	/VS	:	GREEN:	O,A	Video-Signal grün
05-06	HS	/HS	:	BLUE:	O,A	Video-Signal blau
07-08	GND	RED	:			
09-10	GND	GREEN	:	VS,HS:	O,OD	pos. Sync-Signale
11-12	GND	BLUE	:	/VS,/HS:	O,OD	neg. Sync-Signale
13-14	GND	CKPIX	:			
15-16	GND	--	:	CKPIX:	I/O	Bildpunkttakt

N3: Kaskadierungssignale für Erweiterung

				Leitung	Typ	Funktion
01-02	+5V	+5V	:	+5V,+12V	O,U	Versorgungsspannungen
03-04	GND	R0	:			für externe Geräte
05-06	GND	R1	:	GND:	O,U	gemeinsame Masse
07-08	GND	R2	:			
09-10	GND	R3	:			
11-12	GND	G0	:	R0-3:	I/O	Rotsignal (4 Bit)
13-14	GND	G1	:	G0-3:	I/O	Grünsignal (4 Bit)
15-16	GND	G2	:	B0-3:	I/O	Blausignal (4 Bit)
19-20	GND	G3	:			
17-18	GND	B0	:			
21-22	GND	B1	:	HS,VS:	O,OD	pos. Sync-Signale
23-24	GND	B2	:	/HS,/VS:	O,OD	neg. Sync-Signale
25-26	GND	B3	:			
27-28	VS	/VS	:			
29-30	HS	/HS	:			
31-32	--	--	:			
33-34	+12V	CKPIX	:	CKPIX:	I/O	Bildpunkttakt

----- GRIP-COLOR Stückliste -----

----- Widerstände -----

R1-4,7,8	470 Ohm	
R5	68 Ohm	
R6	150 Ohm	
R9,10,12, 14,15,17, 19,20,23, 24,28,30	360 Ohm	
R11,13,16, 18,21,22, 25,26,27, 29,31,32	180 Ohm	
R33-35	1 kOhm	
R36-38	68 Ohm	
P1-3	100 Ohm	Trimmer

----- Kondensatoren -----

C1-19	100 nF	Keramik
C20	10 uF	Tantal

----- IC's ----------

Z1	74HC/LS245	Transceiver
Z2	74HC373	Latch
Z3	74HC/LS02	4xNOR
Z4,14	74HC/LS00	4xNAND
Z5	74HC/LS32	4xOR
Z6	74HC/LS08	4xAND
Z7	74HC/LS139	Dekoder
Z8	74HC/LS240	Buffer 8 Bit
Z9,10	74HC/LS258	Multiplexer
Z11-13	74HC/LS166	Shift-Register
Z15-23,26-34,36-41	4164-15	RAM 64Kx1
Z35	74HC/LS00	4xNAND
Z25	74HC/LS273	Latch 8 Bit
Z42	74HC/LS158	Multiplexer
Z43-45	74S189	RAM 16x4
Z46	74HC/LS74	Flip-Flop
Z24	74HC/LS 05	

----- Sockel -----

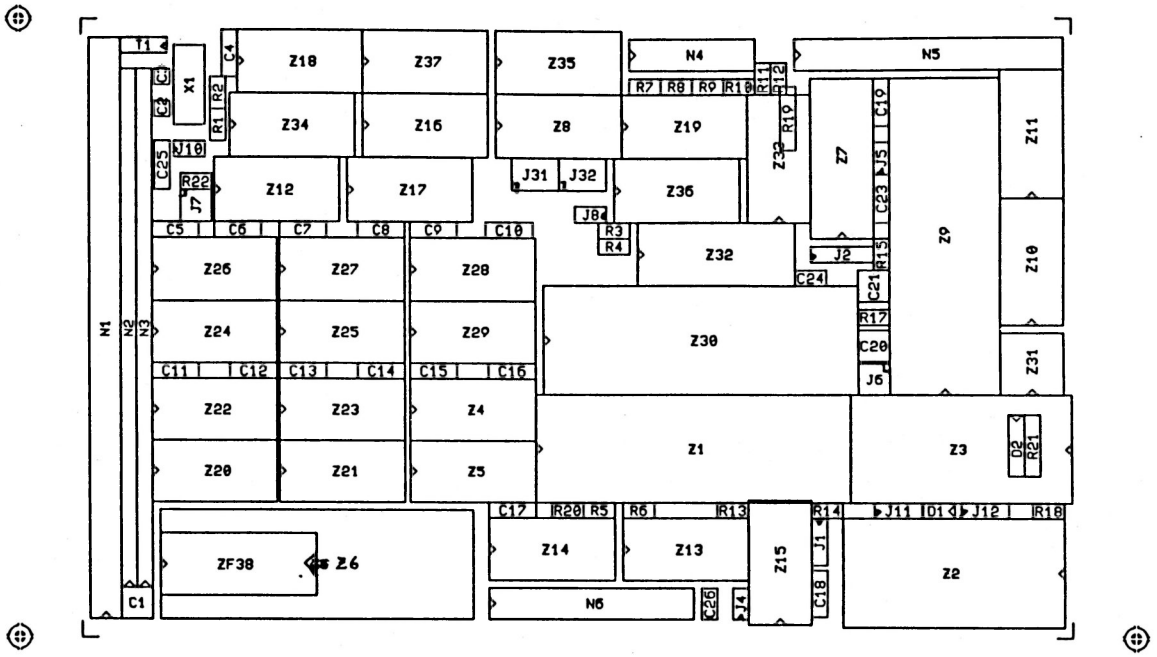
7x	14pol.
34x	16pol.
4x	20pol.

----- Stecker -----

N1	VG 96pol.
N2	Pfosten 8x2
N3	Pfosten 17x2

----- Sonstiges --------

T1-3	BC547 o.ä.



GRIP 4/1 09-Mar-86 GR4
ema30 BPL=1

GRIP Stückliste

ACHTUNG - Bei etwaigen Widersprüchen zwischen Schaltplan, Bestückungsplan und Stückliste ist immer diese Stückliste maßgeblich!

-- Widerstände

R1,2,6	270 Ohm
R3-5	2.2 kOhm
R7	150 Ohm
R8	68 Ohm
R9	470 Ohm (BAS-Signalpegel)
R10-12	470 Ohm
R14,16,18	2.2 kOhm
R13,15	10 kOhm
R17	180 Ohm (Lautstärkestufen)
R19	entfällt
R20	2.2 kOhm
R22	2.2 kOhm (Nur für Akku)
R21	150 Ohm
TR1	2.2 kOhm (Lichtgriffel-Pegel)

-- Kondensatoren

C1	10 uF/6.3 V	Tantal
C4-C19	100 nF	Keramik
C2,C3	4.7 uF/22 V	Tantal
C20	10 uF/22 V	"
C21	6.8 uF/6.3 V	"

-- IC's

Z1	Z8400A	Z80A-CPU
Z6	8255	I/O-Port
Z9	MK3801-4	Z80A-STI
Z30	MM6845	VDC (Motorola)
Z2	27256-25	32KByte-EPROM
Z3	6264-15	8KByte-RAM
Z20-27	4164-15	64KBit-DRAM
Z18	74S04	6x Inverter
Z19	7406/S05	6x Inverter o.K.
Z16	74LS20	2x NAND, je 4 Eing.
Z17	74LS27	3x NOR, je 3 Eing.
Z15	74LS32	4x OR, je 2 Eing.
Z35	74LS74	2x D-Flipflop
Z12	74LS138	Dekoder 8 aus 3
Z13	74LS139	2x Dekoder 4 aus 2
Z34	74LS163	Zähler 4 Bit
Z36	74LS166	Schieberegister
Z33	74LS173	Latch 4 Bit
Z32	74LS244	Buffer Tristate
(Z38)	74LS245	Bus-Buffer (optional)
Z37	74HC/LS32	4x OR
Z4,5,28,29	74HC258	MUX Tristate inv.
Z8	74HC259	DEMUX Latch
Z14	74LS368	Buffer Tristate
Z7	74LS374	Latch 8 Bit
Z11	1489	V24-Empfänger
Z10	1488	V24-Treiber
Z31	NE555	Timer

-- Sonstiges

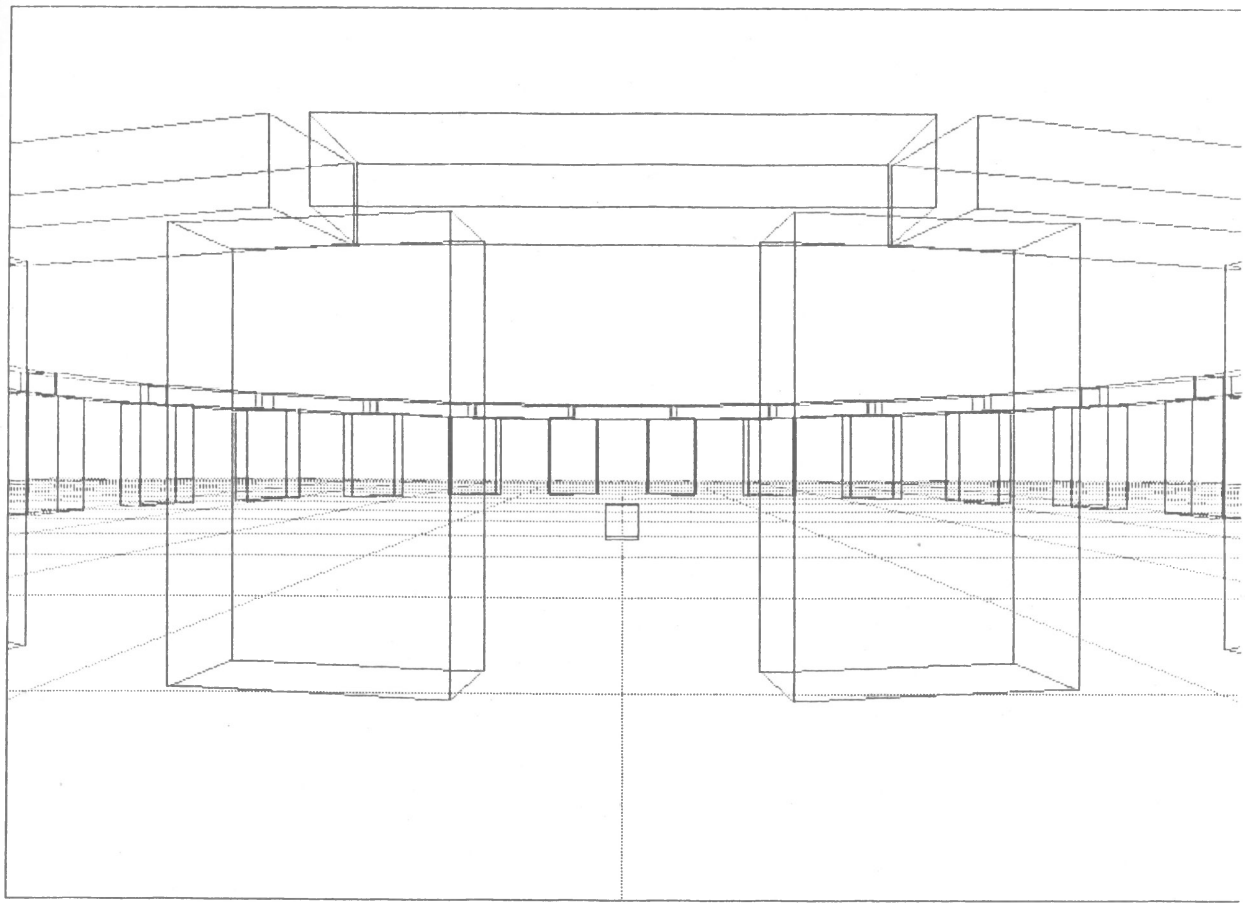
X1	15 MHz-Quarz
D1	LED grün
D2	AA118 (oder Drahtbrücke)
J-12	div. Jumper

-- Sockel

1x	8-pol.
8x	14-pol.
19x	16-pol.
2x	20-pol.
1x	24-pol.
1x	28-pol.
4x	40-pol.

-- Stecker

N1	96pol.	VG-Leiste
N2	16pol.	Pfostenleiste
N3	34pol.	"
N4	26pol.	"



Hardcopy von GRIP

Unterschiede GRIP-5.53/4.53 zu GRIP-4.1

10.10.1987 Johannes Christian Lotter

1. Geschwindigkeitsoptimierung: Der Bildaufbau (Zeichengenerierung) ist um etwa 30% schneller. Das Zeichnen horizontaler Vektoren erfolgt 10x schneller, das Zeichnen vertikaler Vektoren 2x schneller als in jeder anderen Richtung. Das Zeilenlöschen (Aufwärtsrollen unter WORDSTAR oder TURBO-PASCAL) ist je nach Cursorposition bis zu 5x schneller als früher.

2. Folgende Parameter sind jetzt zusätzlich resident gespeichert:

   - Interlaced-Modus ‹ESC› ‹ESC› 'r' a
   - Monitorwerte ‹ESC› ‹ESC› '9' x
   - Zeilenabstand/Smooth Scroll ‹ESC› ‹ESC› '6' x
   - Zeichensatz-Nationalität ‹ESC› ‹ESC› '7' a

   Für Sonder-Anwendungen läßt sich vom Hersteller die Default-Einstellung (Statuszeilen, Interlaced-Modus, Format etc.) leicht modifizieren.

3. Ab GRIP-5 werden 128 KByte Bildspeicher unterstützt. Dazu wurde der Befehl ‹ESC› ‹ESC› '5' a um einen Parameter erweitert:

   a = '6': Video-RAM High-Page umschalten.

   Danach steht ein zweiter Bildschirmspeicherbereich zur Verfügung (auch im Interlaced-Modus). Dieser Befehl ist nur wirksam, wenn GRIP mit dynamischen RAM's vom Typ 41256 bestückt und J8 nicht gebrückt ist.
   Es sollte sichergestellt sein, daß vor Ausführung des Befehls der Druckerspooler leer oder abgeschaltet ist.

4. Betriebsart: Die Befehle ‹ESC› ‹ESC› 'r' a zum Umschalten des Interlaced- bzw. Non-Interlaced-Modus führen keinen Reset mehr aus, sondern löschen nur noch den Bildschirm. Die anderen Daten bleiben unverändert. Der Host-Puffer wird nicht gelöscht, die Verzögerungszeit ist nicht mehr erforderlich.

5. Der Befehl ‹ESC› ‹ESC› '6' a wurde bei GRIP-5 erweitert:

   a = !0!s1!s0!d!z3!z2!z1!z0! (Bits 7-0, 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

   Mit der neuen 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.
Der Befehl ‹ESC› ‹ESC› '6' ist im Interlaced-Modus inaktiv (Zeichenhöhe dort immer 16 Pixel). Für das Vektorzeichnen (Tektronix-Modus non-interlaced) wird automatisch eine Zeichenhöhe von 8 Pixeln eingestellt.

6. Smooth Scroll (pixelweise gleitendes Rollen des Bildschirms) ist bei GRIP-5 implementiert. Dabei 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 'sØ'-Bit mit ‹ESC› ‹ESC› '6' (s. o.) abschalten. Er ist im Interlaced-Modus inaktiv.

7. Double Size (doppelte Zeichenhöhe für den gesamten Bildschirm) ist ab GRIP-5 für den Non- bzw. Pseudo-Interlaced-Modus implementiert und läßt sich über das d-Bit mit ‹ESC› ‹ESC› '6' (s.o.) einschalten. Durch die doppelte Zeichenhöhe verringert sich die Anzahl der verfügbaren Zeilen auf die Hälfte.

8. Bei GRIP-5 ergibt sich durch das verbesserte Sync-Timing des Controllers auf guten Monitoren (z.b. CRT-201 von CONITEC) ohne Verzerrung der oberen Zeile eine vertikale Auflösung von 576 Pixeln, was 36 Zeilen bei einer 8x8-Zeichenmatrix entspricht. Daher lassen sich mit ‹ESC› ‹ESC› '8' jetzt 36 Zeilen einstellen (bzw. 41 Zeilen bei 7 Pixel Zeichenhöhe).
Um zur GRIP-4 kompatibel zu bleiben, beginnt der Tektronix-Koordinatenursprung bei GRIP-5 wie vorher 560 Punkte unter dem oberen Bildschirmrand. Der Ursprung läßt sich jedoch mit der neuen Tektronix-Sequenz

‹ESC› 'A' a
a = 20H + Y-Koordinatenbereich/16 (default: 'C' (43H))

16-pixelweise nach oben oder unten verschieben. Beispiel: ‹ESC› 'A' 'D' setzt den Ursprung ganz nach unten, so daß die vollen 576 Pixelzeilen ausgenutzt werden. Der Ursprungswert ist resident abgespeichert.

9. Tektronix-Alpha-Modus: Der Cursor läßt sich mit dem Befehl ^J (ØAh) abwärts, mit ^K (ØBh) aufwärts, mit ^H (Ø8h) nach links und mit ^L (ØCh) nach rechts bewegen. Der Cursor selbst ist im Tektronix-Modus nicht sichtbar. Die Alpha-Zeichen werden nicht mehr 'zeilenbündig' positioniert, sondern können vertikal auf beliebige Pixelpositionen geschrieben werden.

10. REPLACE-Modus: Zusätzlich zu den Vektor-Modi 'Write' (^S), 'Erase' (^Q) und 'Invert' (^R) wurde ein neuer Modus REPLACE (^P) eingebaut, bei dem die 1-Bits der Maske mit der Vordergrundfarbe und zugleich die Ø-Bits mit der Hintergrundfarbe geschrieben werden. Der REPLACE-Modus verlangsamt die Vektor-Zeichengeschwindigkeit um etwa 10%.

11. Flächenfüllen (<ESC> 'F') ist implementiert (nur für konvexe Flächen). Das Füllen erfolgt in dem gewählten Linientyp (Maske, Muster oder Spray - s.u.). Ein beliebiges Polygon, das sich auch überschneiden darf, läßt sich mit <ESC> 'F' füllen, indem innerhalb jedes zweiten Eckpunktes das Flächenfüllen neu gestartet wird. Die Füllgeschwindigkeit liegt bei 200.000 Pixel/sec.

12. Linientyp PATTERN (<ESC> 'P'): Alle folgenden Vektoren oder Flächen erhalten eine Musterstruktur, die dem Zeichen 7Fh (<DEL>) entspricht und sich alle 8 Pixels in jeder Richtung wiederholt. <DEL> ist defaultmäßig eine Graufläche, kann aber wie jedes andere Zeichen über Zielkanal 4 umdefiniert werden.

13. Linientyp SPRAY (<ESC> 'S' a): 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. Er war bisher in der 3D-Grafikbibliothek (Best.-Nr 610127) als Userprogramm realisiert und wurde jetzt in die GRIP direkt übernommen.

14. Transfer: Der Befehl <ESC> <ESC> 'f' xx y z.. besitzt einen Schnellübertragungsmodus, der aktiviert wird, wenn nach Empfang der xx-Bytes keine Zeichen mehr im Hostpuffer stehen. Hierzu sollte eine kurze Verzögerungsschleife (100-500 ms) nach Senden der beiden xx-Bytes durchlaufen werden. Der Schnellmodus überträgt Daten ins Video-RAM mit etwa vierfacher Geschwindigkeit gegenüber dem Normal-Modus.

15. Userprogramm: Die Register IX und IY dürfen nicht verändert werden, solange die Interrupts nicht mit 'DI' gesperrt wurden.

16. Hardcopy: Mit <ESC> <ESC> 'h3' läßt sich jetzt der Drucker NEC P7 vorwählen. Die Default-Einstellung ist EPSON. Achtung: Wenn mit ESC 'A' (s.o.) im Tektronix-Modus die Bildhöhe verändert wurde, muß der Initialisierungsstring auf die neue Zeilenzahl eingestellt werden! Voreingestellt sind 560 Zeilen. Genaueres entnehmen Sie bitte Ihrem Druckerhandbuch.

17. Blockfüllen: Im Tektronix-Modus wird mit <ESC> 'B' in den 'Blockfüll-Modus' umgeschaltet. Von nun an werden zwischen den eigegebenen Koordinaten statt Vektoren Rechtecke gezeichnet, die in der aktuellen Farbe bzw. mit dem eingestellten Muster gefüllt sind. Die gesendeten Koordinaten bilden dabei die beiden diagonal gegenüberliegenden Ecken des Rechtecks.
Der Blockfüll-Modus eignet sich auch gut zum schnellen Löschen rechteckiger Bildbereiche. Er wird durch Einschalten des Vektor-, Punkt-, Inkremental- oder Alphamodus wieder abgeschaltet.

18. Dreieckfüllen: Im Tektronix-Modus wird mit <ESC> 'T', gefolgt von drei Koordinaten, ein in der aktuellen Farbe bzw. mit dem eingestellten Muster gefülltes beliebiges Dreieck gezeichnet. Die Koordinaten bilden dabei die Ecken des Dreiecks.
Dieser Befehl läßt sich auch zum Zeichnen beliebiger gefüll-

ter Polygone verwenden, da jedes Polygon aus Dreiecken zusammengesetzt werden kann.

.cp4

19. IBM-Tastatur: GRIP unterstützt jetzt sowohl Tastaturen, die ASCII-Codes aussenden (z.b. PREH PC-1A; J3 in Pos. 10-11), wie auch normale IBM-Tastaturen (PREH PC-1; J3 in Pos. 8-9). In beiden Fällen wird die Datenleitung an **KDATA**, die Taktleitung an **KCLOCK** und **KBF** (Stecker **N6**) angeschlossen. Eventuell, je nach Tastatur, muß an KDATA noch ein Pull-up-Widerstand (2.2 kOhm nach +5V) angeschlossen werden.

Die PREH PC-1A läßt sich auch als serielle 1200-Baud-Tastatur betreiben; zusätzlich kann der Reset-Ausgang dieser Tastatur mit dem /PCL- oder /INIT-Anschluß der Karte verbunden werden.

```
        3 o (Reset) ----------------------> /PCL (nur PREH PC-1A)
          5 o (+5V) ----------------------> +5V
   KBD    2 o (Data) ---------------------> KDATA + Pull-Up 2k2
          4 o (GND) ----------------------> GND
        1 o (Clock) ----------------------> KCLOCK + KBF
```

Umbau GRIP-4 auf GRIP-5

Hierzu ist der Video-Controller Z30 (MM6845) gegen den Typ HD6345 auszutauschen, ebenso - bei Verwendung des zweiten Bildspeicherbereichs - die RAM's Z20-27 gegen die Typen 41256. Das EPROM muß durch die Version 5 ersetzt werden. Der Jumper J8 muß offen sein. Ein Umbausatz, der Controller und EPROM beinhaltet, ist bei CONITEC erhältlich (Best.Nr. 610427).

Routine

Part of GRIP-listing

```
TPAR:   LD      HL,MODUS
        RES     4,(HL)
        CALL    PBIOS
        RET
        ~

TS12I:  LD      HL,TSTAB+(12)
        JR      TSCOM
        ~
```

16 =	4800	BAUD
14 =	2400	BAUD
12 =	1200	BAUD
10 =	600	BAUD
8 =	300	BAUD
6 =	150	BAUD
4 =	110	"
2 =	75	"
0 =	50	"

Change this offset to the offset of the desired baudrate.

Set the Baudrate jumper on the GRIP-card to 1200 baud