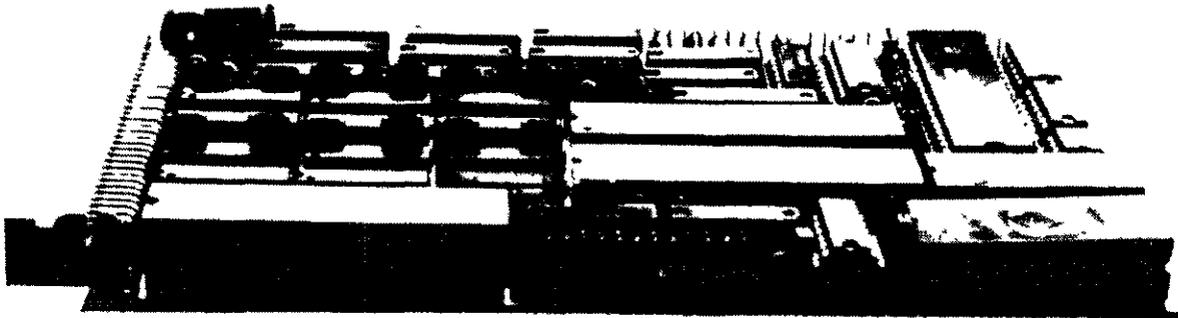


GRIP

Grafik-Interface-Prozessor

Handbuch



 **CONITEC**
J. H. Christian Lotter KG · Datensysteme

GRIP

Grafik-Interface-Prozessor

Handbuch

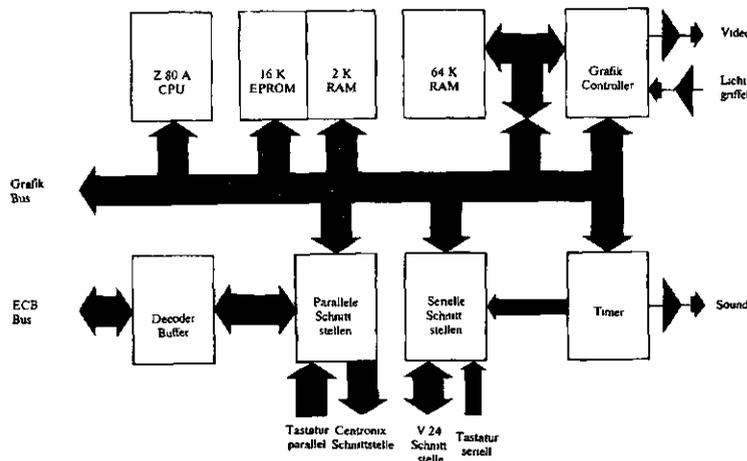
INHALTSVERZEICHNIS

Hardware	1.1
Schaltung	1.3
Schnittstellen	1.3
Kommunikation	1.4
Peripherie	1.5
Video-Logik	1.6
Bildspeicher	1.6
Lichtgriffel	1.7
Externe Synchronisation	1.8
Monitoranschluß	1.8
Aufbau und Inbetriebnahme	1.9
Befehlssatz	2.1
Doppelescape-Sequenzen	2.3
Konfiguration	2.3
Modi	2.7
Datentransfer	2.8
Kanäle	2.10
Farbeinstellung	2.12
Sonstige Befehle	2.14
TVI-Modus	2.16
Control-Codes	2.16
Escape-Sequenzen	2.17
Tektronix-Modus	2.19
Control-Codes	2.19
Escape-Sequenzen	2.20
Vektormodus	2.20
Punktmodus	2.22
Inkrementalmodus	2.22
Beispielprogramm	2.23
Statuszeile	2.24
Kommando-Übersicht	2.25
Adressbelegung	3.1
Jumper	3.5
Programmierung des Video-Controllers	4.1
Programmierung des STI	4.4
Schaltplan	5.1
Bestückungsplan	5.3
Stückliste	5.3
Steckerbelegungen	5.4

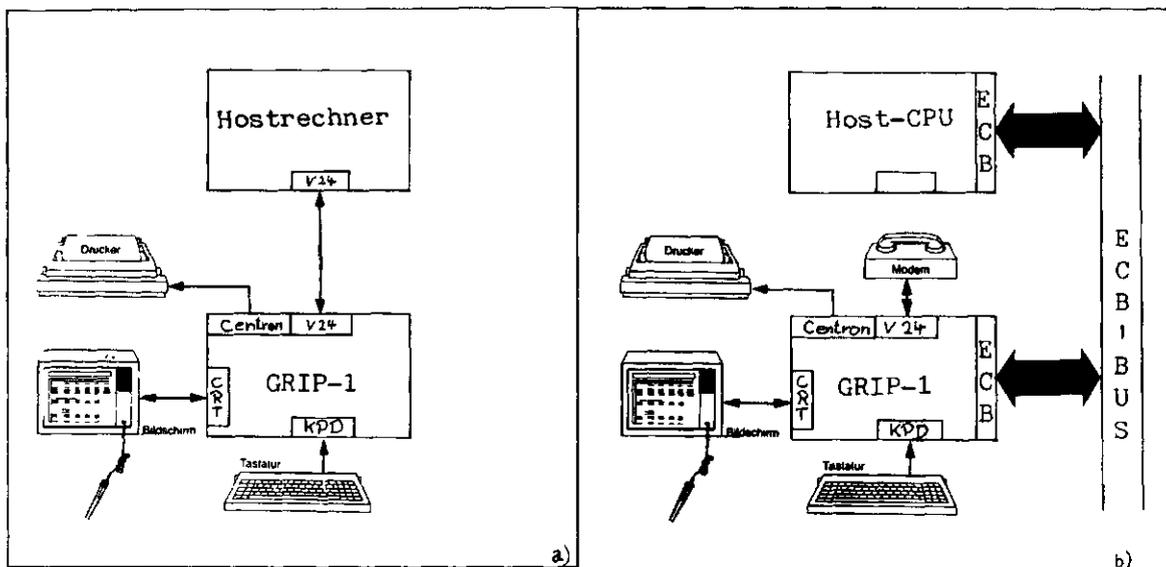
GRIP - Hardware

GRIP vereinigt alle üblichen Ein/Ausgabekanäle eines Rechnersystems auf einer Europakarte. Sie läßt sich wie ein Terminal ansteuern und enthält ein Videointerface für Text und hochauflösende Grafik, mehrere serielle und parallele Schnittstellen zur Verbindung von Hostrechner und Peripherie sowie Anschlüsse für Lichtgriffel und Lautsprecher.

Ein eigener Z80A-Prozessor übernimmt den Datentransfer und die Grafikberechnungen. Das Blockschaltbild gibt einen Überblick über die internen Signalströme.



Die Karte läßt sich auf zwei verschiedene Arten vom Host-Rechner ansteuern (Bild): über eine V24/RS232-Schnittstelle oder direkt über den ECB-Bus. Beim Betrieb zusammen mit der Prozessorkarte PROF-80 wird z.B. von dieser letzten Möglichkeit Gebrauch gemacht.



GRIP - Hardware

Da GRIP 66 KByte RAM enthält, können alle Schnittstellen gepuffert werden, d.h. die eingehenden Daten werden vor der Verarbeitung oder Ausgabe erst einmal zwischengespeichert. Bei der Version GRIP-2 steht etwa die Hälfte des RAM als Pufferspeicher zur Verfügung, der Rest wird für Tabellen, Text und Grafik gebraucht. GRIP-3 benutzt fast das ganze RAM als Grafikspeicher.

Die Grafik verfügt, je nach Software-Version, über eine Auflösung von 768x280 oder 768x560 Bildpunkten. Sie wird durch Vektor-Zeichenbefehle unterstützt. Farbgrafik (16 oder 128 frei definierbare Farben pro Bildpunkt) ist mit der Zusatzkarte GRIP-COLOR möglich.

Davon abgesehen läßt sich die Karte wie ein normales Text-Terminal einsetzen. Sie enthält im Text-Modus 9 verschiedene Zeichensätze, darunter alle griechischen und sonstigen europäischen Sonderzeichen. Bei der Version GRIP-2 können alle Zeichen vom Benutzer selbst umdefiniert werden.

Die Zeichen lassen sich in zwei verschiedenen Größen auf den Bildschirm bringen (7x7 oder 5x5 Bildpunkte innerhalb einer 8x8-Matrix bei GRIP-2; 7x13 oder 5x9 Bildpunkte innerhalb einer 8x16-Matrix bei GRIP-3). Dies ermöglicht hoch- oder tiefgestellte Indizes und mathematische Formeln. Die Zeichen werden ebenso wie die Grafik von der Z80-CPU in's Video-RAM geschrieben.

Zusätzlich sind in der Software noch Sonderfunktionen vorgesehen: Zwei Statuszeilen, eine abfragbare Echtzeituhr, ein Monitortestbild usw. In dem Befehlssatz sind die Steuercodes häufig benutzter Terminals enthalten. Die Text-Steuerbefehle entsprechen denen des Televideo-TVI950, die Grafikbefehle sind die gleichen wie beim Tektronix-Grafikterminal TX4010. Dies erleichtert die Softwareanpassung und wird von den üblichen Softwarepaketen, wie z.B. WORDSTAR, CALCSTAR, DBASE usw. unterstützt.

Zur Dateneingabe gibt es Anschlüsse für einen Lichtgriffel und eine parallele oder serielle ASCII-Tastatur. Über eine interne Tabelle läßt sich jede Taste mit einem beliebigen Befehl oder einer Zeichenfolge belegen, die dann auf Tastendruck zum Hostrechner gesendet wird. Das erhöht in vielen Fällen den Bedienungskomfort ("Funktionstasten"). Damit keine Taste "verlorengehen" kann, werden die letzten 64 Eingaben im Tastaturpuffer zwischengespeichert. Zur visuellen Kontrolle ist ein Teil dieses Puffers in der Statuszeile sichtbar.

Für akustische Meldungen (z.B. um Schlachtenlärm für Computerspiele zu liefern) besitzt GRIP einen NF-Ausgang, der Signale in drei Lautstärke-stufen von einem programmierbaren Ton- und Melodiegenerator liefert. Dieser Ausgang wird auch für das ASCII-Glockensignal (BELL) und für den abschaltbaren "Tastenklick" benutzt. Man darf hier natürlich keine HiFi-Qualität erwarten.

Schließlich gibt es noch ein Druckerinterface mit einem Zeichenpuffer von 30 KByte Länge (nur bei Version GRIP-2). Damit können ca. 15 DIN-A4-Seiten Text auf einen Schlag an die Karte ausgegeben und dann mit größt-

GRIP - Hardware

möglicher Geschwindigkeit gedruckt werden, während der Host-Prozessor bereits wieder für andere Aufgaben frei ist. Dieser sogenannte Spooler-Betrieb ist immer dann sinnvoll, wenn ein schnelles System Daten an ein langsames Peripheriegerät sendet, wie z.B. an einen Drucker oder Plotter.

Schaltung

Der Übersicht halber wurde der Schaltplan in zwei Funktionsgruppen aufgeteilt: CPU-Teil mit Schnittstellen und Grafik-Teil mit dem Video-Controller und dem 64K-RAM.

Ein aus drei Invertern (Z18-A,B,C) gebildeter Quarzoszillator erzeugt den zentralen 16-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 Z80A-CPU (Z1) läuft mit 4 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. Das Betriebssystem und die verschiedenen Zeichensätze stehen in einem 16-KByte-EPROM vom Typ 27128 (Z2); ein statisches 2-KByte-RAM (Z3) bietet Platz für Stack, Parameter und einen Teil der Zeichenpuffer. Dieses Zusatz-RAM ist notwendig, weil die CPU auf den 64-KByte-Hauptspeicher nur zu bestimmten Zeiten (Dunkelphasen) zugreifen kann. Andernfalls gibt es Bildstörungen in Form von schwarzen Streifen auf dem Schirm.

Der Demultiplexer mit Latchausgängen, Z8, steuert acht Leitungen für die Kontrollflags FLASH, PAGE, VOLO-1, CCO-2 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.

Schnittstellen

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 program-

GRIP - Hardware

mierbar. 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.

K o m m u n i k a t i o n

Über Port A kann die System-CPU mit der internen CPU in Verbindung treten. Daten, die von "außen", d.h. vom Bus, in den Port 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.

Vom ECB-Bus sieht der Datentransfer so aus: Auf der Adresse Clh werden die Daten mit I/O-Befehlen eingeschrieben oder ausgelesen, auf C0h der Status abgefragt. Ist Bit 7 des Statusbytes auf C0h auf "1" gesetzt, so steht auf Clh ein Datenbyte von GRIP zum Lesen bereit. Status-Bit 6 signalisiert mit "1", daß das nächste Datenbyte nach Clh geschrieben werden kann.

Zur Verdeutlichung des Ganzen ein Programmbeispiel im Z80-Code zur Ausgabe des Buchstaben "A" auf den Bildschirm:

```
LD C,4lh ;ASCII-Zeichen für "A"
Ll: IN A,(C0h) ;Abfragen des Statusports
BIT 6,A ;Fertig zum Einschreiben?
JR Z,Ll ;Wenn nicht, wieder abfragen
LD A,C ;Auszugebendes Zeichen
OUT (Clh),A ;Einschreiben in den Datenport
```

Für die interne CPU erfolgt der Datentransfer auf die gleiche Weise; allerdings liegt hier der Datenport auf der I/O-Adresse 70h (Port A),

G R I P - H a r d w a r e

das Statusbyte auf 72h (Port C). Eine "1" auf Bit 7 erlaubt das Senden, eine "1" auf Bit 5 das Empfangen des nächsten Bytes. Nach erfolgter Lese- oder Schreiboperation schaltet das entsprechende Statusbit wieder auf "0" um. Zur Beschleunigung des Transfers kann über die STI ein Interrupt ausgelöst werden, sobald eines dieser Bits auf "1" springt.

P e r i p h e r i e

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 mit 5poligem Stecker wird zusätzlich noch der Eingang EXIN als Takteingang 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 VOLO-1 einstellen.

An den SOUND-Ausgang 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. Der Eingangswiderstand muß so abgestimmt werden, daß sich die drei Lautstärkestufen deutlich unterscheiden lassen; notfalls ist R17 zu ändern.

Das Centronix-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. /INIT initialisiert den Drucker und wird beim Einschalten der Karte kurzzeitig aktiviert. /STB 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 das Flag CC2 in den hochohmigen Zustand versetzt werden.

Zwei Ausgänge (CC1-2), ein Eingang (SENSE) und eine bidirektionale

G R I P - H a r d w a r e

I/O-Leitung des STI (EXIN) sind noch für Zusatzfunktionen frei; sie werden von dem Betriebssystem nicht benutzt. Die EXIN-Leitung kann, als Eingang programmiert, auf jeder Flanke des anliegenden Signals einen Vektorinterrupt erzeugen; sie dient alternativ als Takteingang für IBM-Tastaturen.

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.

Zwar sind bereits speziell für Grafik entwickelte Vektorprozessoren erhältlich; sie brauchen aber zuviel Speicher, sind zu langsam oder bieten eine zu geringe horizontale Auflösung, um auch für Text sinnvoll einsetzbar zu sein. Ihr größter Vorteil ist das schnelle Zeichnen von Linien (Vektoren) mit vorgegebenen Anfangs- und Endpunkten. Dies wird hier - entsprechend langsamer - von der Z80-CPU übernommen; der interne Kommando-Zwischenspeicher macht die Zeitverluste allerdings wieder wett.

Die für Textverarbeitung wichtige Übertragung von Zeichen oder Symbolen erfolgt im übrigen mit dem 6845 viel schneller als mit 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 im Text-Modus kaum langsamer als ein reines 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. Das FLASH-Flag entscheidet, ob die CPU oder der Grafik-Controller auf den RAM zugreifen können. Solange der Speicher der CPU zugeordnet ist, bleibt der Bildschirm dunkel.

Die fünf Multiplexer Z4, Z5, Z28, Z29 und Z37 schalten die RAM-Adressen und die Zugriffssignale zwischen CPU und Grafik-Controller um. Für das richtige Timing und das notwendige ständige Auffrischen (Refresh) der Speicherbausteine sorgen die Flipflops Z35-A und Z35-B. 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. Acht aufeinanderfolgende Bytes bilden dabei von oben nach unten die Pixelzeilen des Blocks, beginnend mit Bit 0. 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. Bei der Version GRIP-3, die im Zeilensprung-Verfahren (interlaced) betrieben wird, 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!). Es gibt allerdings Bildschirme mit langer Nachleuchtdauer (mit P39-Phosphorbeschichtung), die für dieses Verfahren geeignet sind. Auch ohne Zeilensprung erzielt man mit nachleuchtenden Monitoren bessere Ergebnisse.

Das echte Zeilensprung-Verfahren ist nur im GRIP-3-Betriebssystem implementiert. Bei GRIP-2 kann ein Pseudo-Zeilensprung eingeschaltet werden, der die vertikale Auflösung durch die Verdopplung jeder Zeile scheinbar erhöht. Dieser Modus sollte aber nur in Verbindung mit den erwähnten P39-Monitoren verwendet werden.

Lichtgriffel

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; Rxx 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

GRIP - Hardware

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 auf einen Eingang des STI geführt, so daß bei Erreichen einer beliebigen Bildschirmposition ein Interrupt ausgelöst werden kann.

Externe Synchronisation

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 entfernt. Die Frequenz sollte ungefähr 15 MHz betragen und läßt sich z.B. mit einer PLL-Schaltung aus dem Videosignal der Kamera gewinnen.

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

Monitoranschluß

An die Karte 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.

Normalerweise werden die Video- und Synchronisationssignale getrennt zugeführt. 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 dazu mit Open-Kollektor-Treibern versehen. Die Pegel können gegebenenfalls durch Ändern der Widerstände R7 und R9 angepaßt werden.

Für spätere Erweiterungen - insbesondere 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 später 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-96/10.

A u f b a u u n d I n b e t r i e b n a h m e

Die Bestückung 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 auf Anhieb funktioniert.

Die Inbetriebnahme sollte trotzdem stufenweise erfolgen, damit eventuelle Fehler sofort lokalisiert werden können. Ein hochohmiger Lautsprecher, ein Multimeter und ein gutes Oszilloskop sind zur Fehlersuche unbedingt erforderlich.

I. Am Anfang steht das Löten. Man sollte mit einem feinen LötKolben 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!

Fast alle später eventuell auftretenden Fehler werden in dieser Phase gemacht. Wenn die Lötarbeit beendet ist, wird es erst richtig spannend: Jetzt werden die Funktionen der Karte der Reihe nach ausgetestet.

II. 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, 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!

III. 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.

IV. Die IC's Z11 (1489) und Z34 (74LS163) werden nun bestückt. An Pin 6 des CPU-Sockels (Z1) sollte ein kräftiges 4-MHz-Taktsignal erkennbar sein, Pin 26 muß auf HIGH liegen.

V. 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 (74LS138, 74LS139, 74LS32). 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 Lötten, zu kontrollieren.

VI. Jetzt kann ein kleiner Lautsprecher oder Kopfhörer an den SOUND-Ausgang angeschlossen werden. Nach Bestücken der IC's Z8 (74LS259), Z10 (1488) und der STI (Z9) muß nach dem Wiedereinschalten ein 1000-Hz-Pfeifton zu hören sein.

VII. 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.

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

IX. In dieser letzten Phase kann nach Bestücken von Z30 endlich der Video-Monitor, wie oben 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 ausprobieren. Jumperstellung (J3) beachten!

GRIP - Befehlsatz

Das 16-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 Schnittstelle auszugeben oder auf dem Bildschirm darzustellen.

GRIP-2 und GRIP-3 unterscheiden sich hauptsächlich in der Auflösung. GRIP-2 arbeitet mit einer 8x8-Zeichenmatrix und 768x280 Bildschirmpunkten; GRIP-3 bietet dagegen eine 8x16-Matrix und eine Auflösung von 768x560 Bildpunkten. Dafür fehlen der Druckerspöler und die Statuszeilen; der User-Zeichensatz umfaßt nur 8 Zeichen.

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 dem Modus, in dem sich die Karte gerade befindet. Im TVI-Modus verhält sich GRIP wie ein TVI950-Terminal, im TX-Modus wie ein Tektronix4010/4014-Grafikterminal mit Speicherbildröhre. Es sind nicht sämtliche, aber alle gebräuchlichen Befehle der beiden Terminals realisiert.

Im TVI-Betrieb wurde das Blinken durch Durchstreichen ersetzt und als zusätzliche Attribute Breit- und Kleinschrift implementiert. Die verschiedenen Edit-Modi sind zu einem einzigen zusammengefaßt. Neu ist die Möglichkeit, verschiedene Zeichensätze gleichzeitig auf den Bildschirm zu bringen und Formeln darzustellen.

Einige hardwarespezifische Befehle (Einstellen der Schnittstelle u.a.) wurden durch Dopplescape-Sequenzen ersetzt. Alle gebräuchlichen Kommandos, die von den Standard-Softwarepaketen (WORDSTAR, DBASE usw.) benutzt werden, sind implementiert.

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

Das Zeilenrollen (Scrollen) geschieht durch Veränderung der Startadresse des Video-Speichers und erfolgt deswegen sehr schnell. Die Ausführung von Befehlen, die Änderungen am Bildschirm bewirken, läuft intern in zwei Schritten ab. Der "Hellprozeß" übernimmt die Daten vom Host und bearbeitet sie für den "Dunkelprozeß". Dieser schreibt sie dann auf den Bildschirm. Um Bildstörungen auszuschließen, darf der Dunkelprozeß nur während der Dunkelphase des Bildschirms, d.h. in den Rücklaufperioden des Elektronenstrahls, aktiv werden.

Deshalb wechseln sich alle 20 Millisekunden der Hell- und der Dunkelprozeß je einmal ab. Die komplizierte Software-Struktur mit zwei Prozessen hat den Vorteil, daß auch bei Zugriff auf den Bildschirm das Bild selbst stabil bleibt. Ohne den Dunkelprozeß würden in dieser Zeit weiße oder schwarze Streifen über den Bildschirm wandern.

GRIP - Befehlsatz

Die meisten internen Datenströme zwischen asynchronen Prozessen laufen über Zwischenspeicher (Puffer), um die Geschwindigkeit der Karte zu erhöhen und Datenverlust auszuschließen. Es gibt Puffer für Befehle (Host-Puffer), für die Kommunikation zwischen Hell- und Dunkelprozeß (Fensterpuffer), für Daten von der Tastatur (Tastaturpuffer), für eine Eingangsschnittstelle (Quellenpuffer) und für den Melodiegenerator (Melodiepuffer). Der mit 30 KByte größte Zwischenspeicher ist der für die Ausgangsschnittstelle (Spooler).

Der Zustand des Tastenpuffers, 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. Bei GRIP-3 gibt es keine Statuszeilen.

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 mit der Bemerkung "COLOR" versehenen Befehle 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-codiertes Byte, z.b. wird "A" durch "41" (34h 31h) codiert
xx = Binäres 16-Bit-Wort (LSB zuerst), z.b. 0Ah 10h = 100Ah = 4106
aaaa = Hex-codiertes 16-Bit-Wort, 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. Ein vorangestellter Pfeil (^) bedeutet ein Control-Zeichen, z.b. <'C> = 03h.

Zeichen in Anführungsstrichen meinen immer ASCII-Zeichen, in Klammern immer Binär-Bytes. Bei einem Binär-Byte werden die einzelnen Bits in der Form

Bit7.Bit6.Bit5.Bit4.Bit3.Bit2.Bit1.Bit0

dargestellt, wenn sie unterschiedliche Bedeutung haben.

GRIP - Befehlssatz

DOPPELESCAPE - SEQUENZEN

Konfiguration

ESC ESC "0" a Spezifiziere V24-Empfängerbaudrate
ESC ESC "1" a Spezifiziere V24-Senderbaudrate
ESC ESC "3" a Spezifiziere serielle Tastatur-Baudrate

a = "0": Baudrate wie J3 (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
 ":" : 9600 Baud (nur V24)
 ";" : 30000 Baud (nur V24)
 "?" : Extern (nur Tastatur, ab Version 2.3)

Beispiel: < ESC > < ESC > "1" "7" (1Bh 1Bh 30h 37h) setzt die Sender-Baudrate auf 1200 Baud.

Im Extern-Modus (<ESC><ESC>"3" "?") kann eine IBM-kompatible Tastatur angeschlossen werden. Dabei ist die Taktleitung mit dem EXIN-Eingang zu verbinden. Eventuell sind Takt- und Datenleitung über Pull-up-Widerstände (4.7 kOhm) auf +5V zu legen.

ESC ESC "2" x Definiere V24-Parameter

x = 0.wl.w0.sl.s0.pl.p0.q

wl-0 = 00: Wortlänge 8 Bit
 01: Wortlänge 7 Bit (def)
 10: Wortlänge 6 Bit
 11: Wortlänge 5 Bit

sl-0 = 01: 1 Stoppbit (def)
 10: 1.5 Stoppbits
 11: 2 Stoppbits

pl-0 = 01: keine Parität (def)
 10: Ungerade Parität
 11: Gerade Parität

q = 0: XON/XOFF-Protokoll aus (def)
 1: XON/XOFF-Protokoll ein

GRIP - Befehlssatz

Beim XON/XOFF-Protokoll sendet die Schnittstelle ein X-OFF-Zeichen (13h), wenn während des Datenempfangs der 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 X-ON-Zeichen (11h) zum Weitersenden aufgefordert.

Bei den Baudraten ab 9600 Baud oder bei Grafik-Betrieb (Vektorzeichen) muß mit RTS/CTS-Handshake oder mit XON/XOFF-Protokoll gearbeitet werden. XON/XOFF-Betrieb ist dann erforderlich, wenn der Host-rechner 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.

ESC ESC "4" x Definiere Tastatur-Kanal und -Parameter

x = 0.w.i.c.0.0.kl.k0

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

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

c = 0: Paralleler Tastatureingang
1: Serieller Tastatureingang

kl-0 = 00: kein Tastenklick (def)
01: leiser Tastenklick
10: normaler Tastenklick
11: lauter Tastenklick

Der Default-Tastaturkanal wird über Jumper J3-2 eingestellt. Die Tastatur ist mit 64 Byte Zwischenspeicher gepuffert. Der "Tastenklick" ist ein kurzer Ton, der bei Betätigung einer Taste zur akustischen Rückmeldung ertönt.

Beispiel: <ESC><ESC> "4" "2" (1Bh 1Bh 34h 32h) paßt GRIP an eine invertierte serielle Tastatur an und setzt die Tastenklick-Lautstärke auf "normal".

ESC ESC "5" a Spooler ein/ausschalten (nur GRIP-2). Der Spooler faßt 30 KByte und dient der Zwischenspeicherung der Daten vom Host zu einem externen Zielkanal.

a = "0": Spooler ein
"3": Kein Spoolerbetrieb (def)

GRIP - Befehlsatz

Sonderzeichen	23h	40h	5Bh	5Ch	5Dh	5Eh	60h	7Bh	7Ch	7Dh	7Eh
ASCII	#	@	[/]	^	\	{		}	~
Britisch	£	@	[/]	^	\	{		}	~
Deutsch	#	§	Ä	Ö	Ü	^	\	ä	ö	ü	ß
Dänisch	#	@	Æ	Φ	Å	^	\	æ	ø	å	~
Französisch	£	à	o	ç	§	^	\	é	ù	è	..
Schwedisch	#	É	Ä	Ö	Å	Ü	è	ä	ö	å	ü
Italienisch	#	§	o	ç	é	^	ù	à	ò	è	ì
Spanisch	#	@	i	Ñ	í	^	\	..	ñ	}	~

ESC ESC "8" a b

Textformat wählen (default: 80x24)

a = Anzahl der Zeilen + 20h

b = Anzahl der Spalten + 20h

Beispiel: <ESC><ESC>"8" ":" "u" (1Bh 1Bh 38h 3Ch 75h) wählt ein Format von 27 Zeilen zu je 86 Zeichen. Das maximal einstellbare Format ist 96x33 (bei GRIP-3: 96x34). Bei Breitschrift ist die Spaltenzahl halbiert. Bei unüblichen Formaten müssen einige Softwarepakete angepaßt werden, was aber z.B. bei WORDSTAR kein Problem ist (Zeilen- und Spaltenzahl auf Adresse 0248h patchen!).

GRIP - Befehlsatz

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

x = 0.i.wl.w0.hl.h0.vl.v0

i = 0: Non-Interlaced (def)
1: Pseudo-Interlaced (nur GRIP-2)

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

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

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

Der Pseudo-Interlaced-Modus verdoppelt scheinbar die vertikale Auflösung, ist aber nur für nachleuchtende Monitore (P39-Phosphor) zu empfehlen. GRIP-3 arbeitet sowieso immer im Interlaced-Modus. Wenn die Zeichen in der obersten Zeile verzerrt werden, ist das Bild nach unten zu verschieben (vl-0 = 10 oder 11).

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

M o d i

ESC ESC "A" TVI-Grundmodus einschalten (def). Alle Zeichen werden in der Vordergrundfarbe geschrieben.

ESC ESC "a" TVI-Farbmodus einschalten (COLOR). Alle Zeichen werden in der mit <ESC> <ESC> "V" eingestellten Farbe geschrieben. In diesem Modus erfolgt die Zeichenübertragung um 50% langsamer!

ESC ESC "T" Tektronix-Grundmodus einschalten. Vektoren und Zeichen werden in die Grundebene geschrieben und erscheinen in der Vordergrundfarbe.

ESC ESC "t" Wie <ESC> <ESC> "T", aber die Statuszeilen sind abgeschaltet.

GRIP - Befehlsatz

- ESC ESC "X" Tektronix-Farbmodus einschalten (COLOR). Vektoren und Zeichen werden in der mit <ESC><ESC>"V" eingestellten Farbe geschrieben. Außerdem erfolgt die Vektorgenerierung wesentlich schneller als im Grundmodus.
- ESC ESC "x" Wie <ESC><ESC>"X", aber die Statuszeilen sind abgeschaltet.

Es wird empfohlen, im TVI-Betrieb den Grundmodus, im Tektronix-Betrieb den Farbmodus zu wählen. Dann ist auch die Übertragungsgeschwindigkeit maximal. Ohne Farbzusatz muß natürlich in beiden Fällen der Grundmodus eingestellt sein.

Datentransfer

Die Datenübertragung zu bzw. von Schnittstellen oder internen Tabellen erfolgt in zwei Schritten: Zunächst muß der Ziel- oder Quellenkanal spezifiziert werden, danach erfolgt der eigentliche Transfer durch einen Übertragungsbefehl. Der einmal gewählte Kanal bleibt bis zur nächsten Änderung eingestellt.

- ESC ESC "C" a Zielkanal einstellen.
- a = "0": Centronics (def)
 "1": ECB-Bus
 "2": V24-Schnittstelle
 "3": Video-RAM
 "4": Zeichensatz
 "5": Userprogramm
 "6": Tasten-Uncodetabelle
 "7": Melodiegenerator

- ESC ESC "c" a Quellenkanal einstellen.
- x = "0": ECB-Bus
 "1": V24-Schnittstelle
 "2": Serieller Tastatureingang
 "3": Paralleler Tastatureingang
 "4": Video-RAM

Der Default-Quellenkanal ist Kanal 1, falls GRIP vom ECB-Bus aus betrieben wird (J3-1 offen); ansonsten Kanal 0. Die Kanäle 2 und 3 können nur benutzt werden, wenn sie nicht bereits als Tastaturkanal definiert sind. Der Quellenkanal ist gepuffert.

- ESC ESC "P" Zielkanal initialisieren (Userprogramm, Unicode-tabelle oder Melodiegenerator). Dieser Befehl löscht den Spooler.

GRIP - Befehlssatz

ESC ESC "I" aaaa Setze Video-RAM-Adresse im Hex-Format.
ESC ESC "i" xx Setze Video-RAM-Adresse binär.

Die Adresse bezieht sich auf direktes Einschreiben/Auslesen in's Video- oder Farb-RAM und wird nach jedem übertragenen Byte hochgezählt. Bit 15 legt fest, in welche Page geschrieben wird. Default = 0000h.

Beispiel: <ESC><ESC> "I" "4000" (1Bh 1Bh 49h 34h 30h 30h 30h) setzt die Adresse auf 4000h (Page 0 - Video-RAM).

ESC ESC "J" a Setze zu definierendes Zeichen im aktuellen Zeichensatz. Nach jeweils 8 (GRIP-2) bzw. 16 (GRIP-3) übertragenen Bytes wird das darauffolgende Zeichen definiert. Default = 20h bei GRIP-2 und 4lh bei GRIP-3.

ESC ESC "D" aa Sende ein Byte im Hex-Format zum Zielkanal.
ESC ESC "d" x Sende ein Byte binär zum Zielkanal.

ESC ESC "E" /aa Hole ein Byte im Hex-Format vom Quellenkanal.
ESC ESC "e" /x Hole ein Byte binär vom Quellenkanal.

Sind von Quellenkanal, z.b. von einer Schnittstelle, keine Daten verfügbar, so wird stattdessen in beiden Fällen eine binäre Null (00h) gesendet.

ESC ESC "F" aa.<CR> Byte-Folge im Hex-Format zum Zielkanal senden. Beendet wird die Folge mit CR .

ESC ESC "f" xx y z.. Byte-Folge binär zum Zielkanal senden. xx gibt die Anzahl der Bytes an (1-65535).

Beispiel: Die Befehle

<ESC><ESC> "F" "2B310D" <CR>
(1Bh 1Bh 46h 32h 42h 33h 31h 30h 44h 0Dh)

und

<ESC><ESC> "f" (03h) (00h) "+1"<CR>
(1Bh 1Bh 66h 03h 00h 2Bh 31h 0Dh)

senden jeweils die Zeichenfolge "+1"<CR> zum Zielkanal.

ESC ESC "G" aaaa/bb.<CR> Bytes im Hex-Format vom Quellenkanal holen. aaaa gibt die Anzahl der zu empfangenden Bytes an (1-65535). Sobald die Bytes gesendet wurden oder - bei Schnittstellen - der Quellenpuffer leer ist, wird die Übertragung mit <CR> beendet.

GRIP - Befehlssatz

ESC ESC "g" xx/y z.. Bytes binär vom Quellenkanal holen. xx gibt die Anzahl an und kann den Wert von 1-65535 annehmen. Ist der Quellenpuffer leer, werden Nullen (00h) gesendet.

80h-FFh Direktmodus: Byte mit gelöschtem Bit 7 direkt zum Zielkanal senden.

Beispiel: (ABh B1h B2h B3h 8Dh) sendet die Zeichen "+123"<CR> zum Zielkanal.

Kanäle

Schnittstellen: Übertragung entweder direkt oder über den Spooler. Auch ohne Spoolerbetrieb sind die Schnittstellen mit ca. 64 Bytes gepuffert.

Video-RAM: Während des Datentransfers ist das Bild dunkelgeschaltet. Das eigentliche Bildschirm-RAM inklusive Zeichensatz belegt die unteren 32 KByte. Die Adressen 8000h-8800h werden von internen Tabellen und Zwischenspeichern benutzt, darüber beginnt der Spooler. Wenn der Spooler abgeschaltet wird (mit <ESC><ESC> "5" "3"), kann dieser Speicherbereich z.B. als RAM-Floppy verwendet werden. Bei GRIP-3 wird das gesamte RAM für den Bildschirm benutzt.

Es wird die mit <ESC><ESC> "V" eingestellte Farbe eingeschrieben. Ausgelesen wird die Bit-Ebene, die durch die drei letzten Bits der Farbnummer bestimmt ist.

Zeichensatz: Jedes Zeichen wird bei GRIP-2 durch 8 Bytes definiert, die einen 8x8-Block bilden; bei GRIP-3 sind es 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 markiert einen weißen Punkt.

Beispiel: Zeichen "l" Bit 01234567

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 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. Insgesamt umfaßt der Speicher für den programmierbaren Zeichensatz 96 Zeichen x 8 Bytes = 768 Bytes (GRIP-2) bzw. 8 Zeichen x 16 Bytes = 128 Bytes (GRIP-3).

GRIP - Befehlsatz

Beispiel: Für irgendeine exotische Anwendung soll anstelle des Zeichens "A" (41h) ein Viereck, anstelle von "B" (42h) ein Dreieck im Zeichensatz erscheinen. Zur schnellen Übertragung wird der Direktmodus verwendet:

<ESC><ESC> "C" "4"	(Zielkanal auf Zeichensatz)
<ESC><ESC> "J" "A"	("A" soll undefiniert werden)
80h 80h FEh C2h C2h C2h FEh 80h 80h	(Viereck; danach ist "B" dran)
80h 80h 98h A4h C2h FEh 80h 80h 80h	(Dreieck)

Userprogramm: Es ist Platz für 256 Bytes (GRIP-3: 64 Bytes); das Programm beginnt auf der Adresse 4384h und muß mit einem RETURN-Befehl (C9h) enden. Für den Stack stehen 12 Ebenen zur Verfügung.

Tasten-Umcodetabelle: Für jede gedrückte Taste kann statt des Original-Tastencodes ein beliebiger Zeichen-String zum Host gesendet werden. Die Umcodierung erfolgt über eine Tabelle, die bis zu 256 Bytes (GRIP-3: 128 Bytes) enthalten darf. Das Format ist: Tastencode (1 Byte) - Stringlänge (1 Byte) - String - nächster Tastencode usw. Die Stringlänge 0 signalisiert das Ende der Tabelle.

Beispiel: Aus einer amerikanischen QUERTY-Tastatur soll eine deutsche Tastatur gemacht werden. Dazu müssen die "Y"- und "Z"-Tasten vertauscht werden. Die Tastatur besitzt außerdem eine Funktionstaste, die bei Betätigung den Code 88h sendet. Beim Druck auf diese Taste soll 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> "C" "6"	(Zielkanal = Umcodetabelle)
<ESC><ESC> "P"	(Alte Umcodetabelle löschen)
<ESC><ESC> "F"	(Hex-Übertragung einleiten)
"59015A5A0159"	(Y mit Z vertauschen)
"79017A7A0179"	(y mit z vertauschen)
"19011A1A0119"	(^Y mit ^Z vertauschen)
"88044449520D"	(88h sendet DIR <CR>)
"0000"	(Ende der Tabelle)
<CR>	(Ende der Übertragung)

Melodiegenerator: Es können 128 Töne (GRIP-3: 32 Töne) vor dem Abspielen zwischengespeichert werden. Jeder Ton besteht aus 2 Bytes. Das erste Byte (b1) bestimmt die Tonhöhe F nach folgender Formel:

$$F = (117.2 \text{ kHz}) / (b1) \quad \text{für gerade } b1$$
$$F = (9.375 \text{ kHz}) / (b1) \quad \text{für ungerade } b1$$

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

GRIP - Befehlssatz

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

b2 = 0.l1.l0.d4.d3.d2.d1.d0

l1-0: 00 = Pause
01 = Leiser Ton
10 = Mittlerer Ton
11 = Lauter Ton

d4-0 = Dauer in Einheiten von 20 Millisekunden. 00000 = Melodie-Ende.

Das Abspielen beginnt mit dem Empfang des Melodie-Endes, d.h. eines Tons, dessen zweites Byte auf 00000 endet. Danach wird der Melodiepuffer gelöscht und ist für eine neue Melodie bereit.

Beispiel: Folgende Kommandos lassen eine kleine Melodie ablaufen, die von unserem Software-Virtuosen Bernhard Emese eigens für GRIP komponiert wurde:

<ESC> <ESC> "C" "7"	(Zielkanal = Soundgenerator)
<ESC> <ESC> "P"	(Alten Inhalt vorsichtshalber löschen)
<ESC> <ESC> "F"	(Hex-Übertragung einleiten)
"4230582458045824"	
"50305822681D4622"	
"460D4222420D"	
"0000"	(Ende der Melodie)
<CR>	(Ende der Übertragung)

Farbeinstellung (nur mit GRIP-COLOR)

ESC ESC "V" a Aktuelle Farbnummer einstellen für den Farbmodus.
In dieser Farbe werden von nun an Vektoren und Zeichen geschrieben.

a = Farbnummer (1-63) + 20h
a = " ": Löschen (Hintergrundfarbe)

GRIP - Befehlsatz

ESC ESC "v" a Ebene einstellen zum Auslesen oder Ausdrucken.

a = " ": Grundebene (def)
"0": Farbebene, Bit 0
"1": Farbebene, Bit 1
"2": Farbebene, Bit 2
"3": Farbebene, Bit 3
"4": Farbebene, Bit 4
"5": Farbebene, Bit 5
"^": Grundebene, invertiert
"0": Farbebene, Bit 0, invertiert
"1": Farbebene, Bit 1, invertiert
"2": Farbebene, Bit 2, invertiert
"3": Farbebene, Bit 3, invertiert
"4": Farbebene, Bit 4, invertiert
"5": Farbebene, Bit 5, invertiert

Dieser Befehl ist notwendig für Hardcopy, Flächenfüllen oder zum Auslesen des Video-RAMs. Der Parameter a gibt das Farbnummer-Bit (0-5) an, dessen Bild ausgedruckt werden soll; ein Leerzeichen (20h) selektiert die Grund-(Schwarzweiß-)Ebene.

ESC ESC "W" a r g b Farbton für Reinfarben und Hintergrund einstellen.

ESC ESC "w" a r g b Farbton für Mischfarben und Vordergrund einstellen.

a = Farbnummer (1-63) + 20h
a = " ": Hintergrundfarbe (bei "W")
a = " ": Vordergrundfarbe (bei "w")

r = Rot-Intensität ("0" - "?")
g = Grün-Intensität ("0" - "?")
b = Blau-Intensität ("0" - "?")

Mit "Vordergrundfarbe" ist die Farbe gemeint, in der Zeichen im TVI- oder Tektronix-Grundmodus geschrieben werden. Im Schwarzweißbetrieb ohne Farbkarte ist das die Farbe weiß.

Reinfarben sind die normalen 63 Farben, in denen im Farbmodus über den Hintergrund gezeichnet werden kann. Mischfarben ergeben sich, wenn die Farbe auf dem Vordergrund erscheint. Für jede Farbnummer gibt es also eine Rein- und eine Mischfarbe. Ist nur eine GRIP-COLOR-Karte angeschlossen, so werden von der Farbnummer nur die 3 niedrigsten Bits ausgewertet, so daß nur 7 Rein- und 7 Mischfarben unterschiedlich sind. Die volle Farbskala ergibt sich durch Kaskadierung von zwei COLOR-Karten.

Bei jeder der insgesamt 126 Farben sowie bei Vorder- und Hintergrundfarbe kann der Farbton und die Helligkeit unabhängig über die Rot-, Grün- und Blau-Intensität in je 16 Stufen bestimmt werden. "0" stellt die niedrigste, "8" z.B. eine mittlere, "?" die höchste Intensität ein.

GRIP - Befehlsatz

Durch die Mischung der drei Farbanteile ergeben sich insgesamt 4096 einstellbare Farbtöne für jede der 128 Farben.

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

Sonstige Befehle

ESC ESC "h" a Druckermodus für Hardcopy einstellen.

a = "0": EPSON FX-80
 "1": Gemini-10X
 "2": Siemens PT-88 (default)

ESC ESC "H" Bildschirminhalt zum Hardcopy-Kanal senden.

Text und Grafik werden Pixel für Pixel ausgedruckt; das Drucken erfolgt direkt oder über den Spooler, je nach Zuordnung. Während der Datenübertragung ist der Bildschirm dunkelgeschaltet. Ohne Spooler wird die Grafik schneller ausgedruckt, dafür ist das Bild länger dunkel. Bei Verwendung der COLOR-Karte kann die auszudruckende Bit-Ebene des Bildschirmspeichers mit <ESC><ESC> "v" vorgewählt werden.

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

x = 0.0.nlp.er.sen.la2.lal.la0

nlp = Lichtgriffel nicht ausgelöst
er = ERROR-Eingang
sen = SENSE-Eingang

la2-0: Lichtgriffel-Adressbits innerhalb eines 8x8-Pixelblocks.

Spalte: 0 1 2 3 4 5 6 7

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

a = Blocknummer + 20h, Zeile 0-34
b = Blocknummer + 20h, Spalte 0-95

GRIP - Befehlssatz

Je nach Ansprechgeschwindigkeit des Lichtgriffels kann die Position in horizontaler Richtung etwas verschoben sein. Dies muß durch das Anwendungsprogramm korrigiert werden. Die Blocknummern (0,0) kennzeichnen die obere linke Ecke auf dem Bildschirm; die Statuszeile ist dabei mit einbezogen.

- ESC ESC "M" aa/bb Userprogramm starten. Die Hex-Zahl aa wird beim Aufruf nach Register A übergeben, die Antwort bb wird nach Programmende von A übernommen.
- ESC ESC "m" x Userprogramm starten. x wird beim Aufruf nach Register A übergeben.
- ESC ESC "O" aa bb Direkte Portausgabe (hex). bb wird an den Port aa ausgegeben.
- ESC ESC "o" x y Direkte Portausgabe (binär). y wird an den Port x ausgegeben.
- ESC ESC "Q" Monitor-Testbild einschalten. Das Testbild beinhaltet Einstellmarken und die Zeichensätze.
- ESC ESC "R" Bildschirm und alle Puffer löschen, alle Funktionen auf die Default-Werte zurücksetzen. Einzig die Uhr läuft weiter.
- Achtung: Dieser Befehl löscht auch den Host-Puffer! Darum können bis zur Ausführung keine weiteren Daten empfangen werden. Es empfiehlt sich, nach Senden von ESC ESC "R" eine kurze Verzögerungszeit einzubauen.
- ESC ESC "S" Statuszeilen umschalten: System-Statuszeile oben ein, User-Statuszeile unten aus (def).
- ESC ESC "s" Statuszeilen umschalten: User-Statuszeile unten ein, System-Statuszeile oben aus.
- ESC ESC "b" a b.<CR> Untere Statuszeile mit Text überschreiben (max. 96 Zeichen). Befehl wird abgeschlossen mit >CR<.
- Bei GRIP-3 gibt es keine Statuszeilen; die Doppelscape-Sequenzen "S", "s" und "b" sind unwirksam.

GRIP - Befehlssatz

ESC ESC "U" a b c Uhr stellen und starten.
ESC ESC "u" /a b c Uhr abfragen, Antwort wie beim Stellen.

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

Beispiel: <ESC><ESC>"U" "C" "7" "P" (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.

TVI - MODUS

Softwarepakete mit TVI950-Installation wie WORDSTAR können im TVI-Modus die Karte ohne weitere Anpassung ansteuern. Das Textformat läßt sich mit <ESC><ESC>"8" ändern.

TVI - Control - Codes

- ^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 (Line Feed).
- ^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 Line Feed, d.h. das Bild rückt eine Zeile nach oben.
- ^J (0Ah) Zeilenvorschub (Line feed). Bewirkt das gleiche wie ^V.
- ^M (0Dh) Wagenrücklauf (Carriage Return). Setzt den Cursor an den Zeilenanfang.
- ^_ (0Fh) Neue Zeile. Entspricht einem Wagenrücklauf plus Zeilenvorschub und setzt den Cursor an den Anfang der nächsten Zeile.

GRIP - Befehlssatz

^^ (1Eh)	Cursor in die Home-Position (Zeile 0, Spalte 0).
^Z (1Ah)	Bildschirm löschen.
^L (0Ch)	Bildschirm löschen, Cursor in die Home-Position. Dieser Code wird für CP/M 3.0 benötigt.
^G (07h)	Glockensignal (Bell).

TVI - Escape - Sequenzen

ESC "+"	Löschen wie <L>.
ESC ";"	Löschen wie <L>.
ESC "." x	Cursorattribute setzen.
	x = "0": Unsichtbarer Cursor
	"1": Blinkender Block (def)
	"2": Stehender Block
	"3": Blinkende Linie
	"4": Stehende Linie
ESC "=" a b	Cursorposition setzen.
ESC "?" /a b <CR>	Cursorposition abfragen.

a = Zeilennummer + 20h
b = Spaltennummer + 20h

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

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 "t"	Wie <ESC> "T".
ESC "Y"	Seite ab einschl. Cursorposition bis zum Ende löschen.
ESC "y"	Wie <ESC> "Y".

GRIP - Befehlsatz

ESC "G" x Zeichenattribute setzen.

x = 0.b.sl.s0.u.i.d.v

sl-0 = 00: Index tief (Subscript)
01: Kleinschrift (Index mittel)
10: Index hoch (Superscript)
11: Normale Schrift (def)

b = Breitschrift (nur GRIP-2)
u = Unterstreichen
i = Invertieren
d = Durchstreichen
v = Unsichtbare Schrift

Von jedem der neun internen Zeichensätze gibt es zwei Schriftarten: Normalschrift und Kleinschrift (Indexschrift). In Wirklichkeit sind also 18 Zeichensätze im EPROM gespeichert. Wegen der teilweisen Redundanz werden dafür aber nur ca. 4 KByte benötigt.

Zusätzlich zu den zwei Schriftarten lassen sich zu jedem Zeichen noch weitere Attribute auswählen: Breitschrift (jedes Zeichen wird in doppelter Breite dargestellt), Invertieren (schwarzes Zeichen auf weißem Feld), durchstreichen und unterstreichen. Alle Attribute lassen sich beliebig kombinieren.

Beispiel: <ESC> "G" "ö" schaltet durchgestrichene, breite, invertierte Kleinschrift ein.

GRIP-2 überträgt Zeichen mit oder ohne Attribute gleich schnell (Ausnahme: Breitschrift). Bei GRIP-3 ist jedes Attribut mit einer Reduzierung der Übertragungsgeschwindigkeit verbunden.

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

Diese beiden Extra-Befehle lassen die übrigen Attribute unverändert. Das Invertieren erfolgt hier nicht wie bei <ESC> "G" durch "Umbau" des gesamten Zeichensatzes, sondern durch Wechseln des Zeichengenerator-Programms. Daher ist die Ausführung viel schneller, die Invers-Zeichengenerierung dafür viel langsamer. Dieser Befehl sollte für die Invertierungs-Option von WORDSTAR verwendet werden.

ESC "n" Bildschirm ein (def).
ESC "o" Bildschirm dunkel.

Dieser Befehl kann 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.

GRIP - Befehlsatz

ESC "\$" Strichgrafik ein.
ESC "%" Strichgrafik aus (def).

Es stehen 15 Strichgrafik-Zeichen zur Verfügung. Diese Zeichen sind eigentlich nicht nötig, denn im Vektormodus lassen sich viel bessere Grafiken erstellen; sie wurden aber implementiert, um so weit wie möglich TVI-kompatibel zu bleiben. Auch andere Programme - z.B. Tabellenkalkulationsprogramme - benötigen manchmal eine solche Strichgrafik.

TEKTRONIX - MODUS

Das Tektronix-Terminal kann selbst in unterschiedlichen Modi arbeiten. Im Alpha-Modus verhält es sich wie ein "abgemagertes" Text-Terminal. Scrollen, Cursorpositionieren und Editieren sind nicht möglich, darum sollte zur Textverarbeitung der TVI-Modus verwendet werden.

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

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.

Die Stärke des Tektronix-Terminals liegt in seiner Grafikfähigkeit. Im Punkt- oder Vektormodus zeichnet es Punkte bzw. Linien zwischen vorgegebenen Bildschirmkoordinaten. Im Inkremental-Modus können kleinere Figuren aus Einzelpunkten schrittweise zusammengesetzt werden.

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.

^Ö (1Ch) Punktmodus einschalten.

^Ü (1Dh) Vektormodus einschalten. Der nächste Vektor wird unterdrückt.

^G (07h) Sonderfunktion nach ^Ü: Der nächste Vektor wird doch gezeichnet.

^^ (1Eh) Inkrementalmodus einschalten.

GRIP - Befehlsatz

- ^_** (1Fh) Alphamodus einschalten. Der Alpha-Cursor wird auf die zuletzt eingegebene Grafik-Koordinate gesetzt.
- ^Q** (11h) Zeichne in der Hintergrundfarbe (Löschen).
- ^R** (12h) Zeichne durch Invertieren.
- ^S** (13h) Zeichne in der Vordergrundfarbe (weiß, default).

Die letzten drei Befehle sind im Farbmodus unwirksam. Hier wird durch Einstellen der Hintergrundfarbe (mit <ESC> <ESC> "V" " ") gelöscht. Ein - in welcher Farbe auch immer - gezeichneter Vektor überdeckt immer die darunterliegenden früher gezeichneten.

Tektronix - Escape - Sequenzen

- ESC <^L> Löscht den Bildschirm, schaltet den Alpha-Modus ein und setzt den Cursor in die obere linke Ecke.
- ESC "G" x Zeichenattribute setzen, wie im TVI-Modus.
- ESC "^"
- ESC "a" Zeichne durchgehende Linien.
- ESC "b" Zeichne gepunktete Linien.
- ESC "c" Zeichne Strichpunkt-Linien.
- ESC "d" Zeichne kurz gestrichelte Linien.
- ESC "d" Zeichne lang gestrichelte Linien.

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. Bei GRIP-3 entspricht jeder Koordinatenpunkt einem realen Bildschirmpunkt, während GRIP-2 gerade und ungerade Y-Koordinaten (vertikal) auf den gleichen Bildpunkt abbildet. Der Nullpunkt liegt immer in der linken unteren Ecke. Es ist darauf zu achten, daß das Bildfenster nicht überschritten wird.

Eine vollständige Koordinate besteht aus 10 Bit in X-Richtung und 10 Bit in Y-Richtung. Sie wird durch bis zu 4 ASCII-Zeichen übermittelt, bei denen jeweils die unteren 5 Bit zur Koordinate beitragen. Die oberen Bits geben den Koordinatenteil 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.

GRIP - Befehlssatz

Beispiel: Die (X,Y)-Koordinate (500,200) schreibt sich binär (01111.10100b,00110.01000b) und wird durch folgende Zeichen kodiert:

hy = "&" (26h); ly = "h" (68h); hx = "/" (2Fh); lx = "T" (54h).

Es läßt sich also eindeutig unterscheiden, zu welchem Koordinatenteil ein Zeichen gehört (bis auf den oberen Teil von X und Y). 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.

Nachdem die erste Koordinate im Vektormodus übertragen wurde, zeichnet GRIP von dort zur nächsten und zu jeder weiteren Koordinate einen Vektor, 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. Die Linienart kann mit den Sequenzen <ESC> "1" - <ESC> "d" bestimmt werden.

Zwei aufeinanderfolgende gleiche Koordinaten erzeugen einen Punkt. Mit <^Ü> (1Dh) läßt sich die Linie zur nächsten Koordinate unterdrücken, erst der übernächste Vektor wird wieder gezeichnet. Die Linienunterdrückung kann jedoch mit <^G> (07h) wieder aufgehoben werden. Dies ist sinnvoll, wenn man den Vektormodus kurz verläßt und dann mit <^Ü> wieder zurückkehrt. Nach Eingabe von ^G^3 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 starren Algorithmus berechnet, der rückwärts im allgemeinen die "Stufen" anderswohin setzt als vorwärts.

GRIP - Befehlssatz

Punktmodus

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

Inkrementalmodus

Im Inkrementalmodus wird von der letzten Grafikordinate 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:

"F"	"D"	"E"	"D" (44h): 1 Punkt nach oben.
			"E" (45h): 1 Punkt diagonal nach rechts oben.
			"A" (41h): 1 Punkt nach rechts.
			"I" (49h): 1 Punkt diagonal nach rechts unten.
"B"	*	"A"	"H" (48h): 1 Punkt nach unten.
			"J" (4Ah): 1 Punkt diagonal nach links unten.
			"B" (42h): 1 Punkt nach links.
"J"	"H"	"I"	"F" (46h): 1 Punkt diagonal nach links oben.

Die Zeichenmodi werden über " ", "P", "Q" und "R" 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 (Transparent).
- "P" (50h): Punkte zeichnen (Vordergrundfarbe).
- "Q" (51h): Punkte löschen (Hintergrundfarbe).
- "R" (52h): Punkte invertieren.

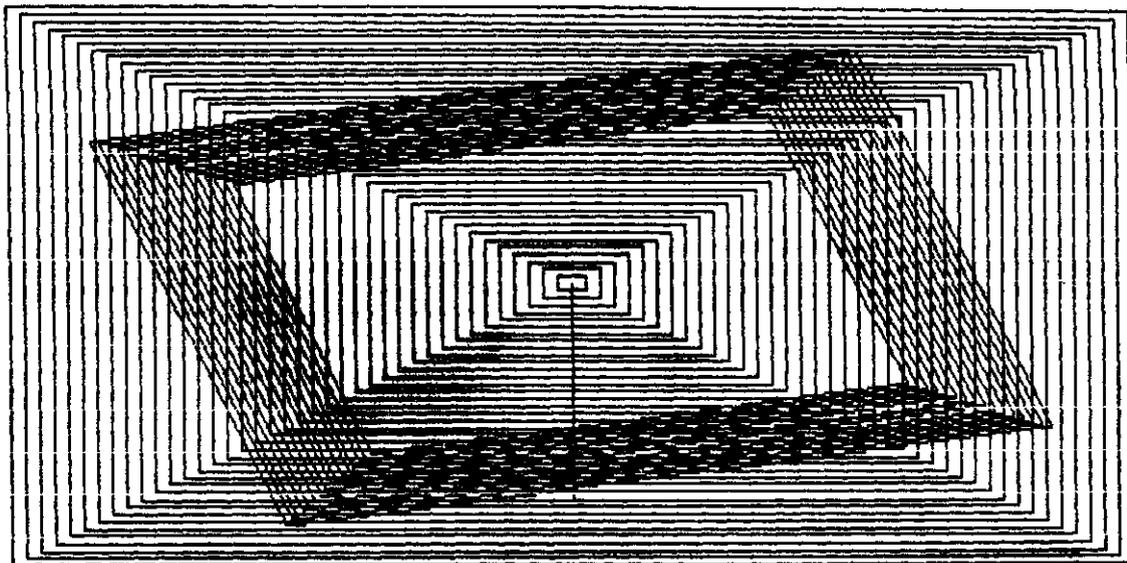
Die "Q"- und "R"-Befehle sind im Original-Tektronix-Befehlssatz nicht enthalten und im Farbmodus unwirksam.

```

10 REM *****
20 REM ***
30 REM *** GRIP-2 Teltronix -Demo mit Hardcopy fuer Epson FX 80 ***
40 REM ***
50 REM *****
60 REM
70 REM
80 REM
90 REM
100 REM verhindere Auftretenfehd
110 WIDTH 255
120 REM Joesche Bildschirm
130 PRINT CHR$(12);
140 REM schalte in Teltronix -Modus
150 PRINT CHR$(27);CHR$(27);"T";
160 REM
170 REM Hauptprogramm
180 REM
190 XOFF=400;YOFF=285
200 XFAI T=300;YFAI T=180
210 FOR I=0 TO 1.2 STEP .02
220 X1=XI;Y1=YI;GOSUB 470
230 NEXT I
240 FOR I=1 TO .7 STEP -.02
250 X1=XI;.6;Y1=YI;.6;GOSUB 470
260 NEXT I
270 REM
280 REM Hardcopy
290 REM
300 REM schalte Spooler ein
310 PRINT CHR$(27);CHR$(27);"S";
320 REM definiere Epson FX-80 als Hardcopy-Drucker
330 PRINT CHR$(27);CHR$(27);"h0";
340 REM Hardcopy
350 PRINT CHR$(27);CHR$(27);"H";
360 REM schalte in TVI-Modus
370 REM
380 REM Ende
390 REM
400 PRINT CHR$(27);CHR$(27);"A";
410 REM positioniere Cursor
420 PRINT CHR$(27);"5 ";
430 END
440 REM
450 REM Diese Unterprogramm zeichnet ein Rechteck auf den Bildschirm
460 REM
470 PRINT CHR$(9);
480 FOR I=1 TO 5
490 X=X1+XFAI T+XOFF;Y=Y1+YFAI T+YOFF;GOSUB 560
500 Z=X1;X1=XI;Y1=Z
510 NEXT I
520 RETURN
530 REM
540 REM Dieses Unterprogramm gibt die X/Y-Koordinaten im Teltronix-Format aus
550 REM
560 Y1Z=Y\32
570 Y2Z=Y MOD 32
580 X1Z=X\32
590 X2Z=X MOD 32
600 X1Z=X1Z+9H20;X2Z=X2Z+9H40
610 Y1Z=Y1Z+9H20;Y2Z=Y2Z+9H60
620 PRINT CHR$(Y1Z);CHR$(Y2Z);CHR$(X1Z);CHR$(X2Z);
630 RETURN

```

GRIP-2.0 H 11";ARUBA 00 010101 I 0211



K o m m a n d o - U 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	Modus	Funktion	Handbuch-Seite
ESC ESC 0	1B 1B 30	a	-	TVI/TX	V24-Empfangerbaurate	2.3
ESC ESC 1	1B 1B 31	a	-	TVI/TX	V24-Senderbaurate	2.3
ESC ESC 2	1B 1B 32	x	-	TVI/TX	V24-Kontrollflags	2.3
ESC ESC 3	1B 1B 33	a	-	TVI/TX	Tastatur-Baudrate	2.3
ESC ESC 4	1B 1B 34	a	-	TVI/TX	Tastatur-Kanal	2.4
ESC ESC 5	1B 1B 35	a	-	TVI/TX	Spooler-Zuordnung	2.4
ESC ESC 7	1B 1B 37	a	-	TVI/TX	Zeichensatz	2.5
ESC ESC 8	1B 1B 38	a b	-	TVI/TX	Bildschirmformat	2.6
ESC ESC 9	1B 1B 39	x	-	TVI/TX	Monitor-Anpassung	2.7
ESC ESC A	1B 1B 41	-	-	TVI/TX	TVI-Grundmodus	2.7
ESC ESC a	1B 1B 61	-	-	TVI/TX	TVI-Farbmodus	2.7
ESC ESC b	1B 1B 62	a b.. CR	-	TVI/TX	Statuszeile schreiben	2.15
ESC ESC c	1B 1B 43	a	-	TVI/TX	Zielkanal	2.8
ESC ESC d	1B 1B 44	aa	-	TVI/TX	Quellenkanal	2.8
ESC ESC e	1B 1B 45	aa	-	TVI/TX	Byte senden	2.9
ESC ESC f	1B 1B 46	aa	-	TVI/TX	Byte empfangen	2.9
ESC ESC g	1B 1B 47	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC h	1B 1B 48	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC i	1B 1B 49	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC j	1B 1B 50	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC k	1B 1B 51	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC l	1B 1B 52	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC m	1B 1B 53	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC n	1B 1B 54	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC o	1B 1B 55	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC p	1B 1B 56	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC q	1B 1B 57	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC r	1B 1B 58	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC s	1B 1B 59	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC t	1B 1B 60	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC u	1B 1B 61	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC v	1B 1B 62	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC w	1B 1B 63	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC x	1B 1B 64	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC y	1B 1B 65	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC z	1B 1B 66	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC A	1B 1B 67	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC B	1B 1B 68	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC C	1B 1B 69	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC D	1B 1B 70	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC E	1B 1B 71	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC F	1B 1B 72	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC G	1B 1B 73	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC H	1B 1B 74	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC I	1B 1B 75	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC J	1B 1B 76	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC K	1B 1B 77	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC L	1B 1B 78	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC M	1B 1B 79	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC N	1B 1B 80	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC O	1B 1B 81	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC P	1B 1B 82	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC Q	1B 1B 83	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC R	1B 1B 84	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC S	1B 1B 85	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC T	1B 1B 86	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC U	1B 1B 87	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC V	1B 1B 88	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC W	1B 1B 89	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC X	1B 1B 90	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC Y	1B 1B 91	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC Z	1B 1B 92	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC [1B 1B 93	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC \	1B 1B 94	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC]	1B 1B 95	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC ^	1B 1B 96	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC _	1B 1B 97	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC `	1B 1B 98	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC a	1B 1B 99	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC b	1B 1B 100	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC c	1B 1B 101	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC d	1B 1B 102	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC e	1B 1B 103	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC f	1B 1B 104	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC g	1B 1B 105	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC h	1B 1B 106	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC i	1B 1B 107	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC j	1B 1B 108	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC k	1B 1B 109	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC l	1B 1B 110	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC m	1B 1B 111	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC n	1B 1B 112	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC o	1B 1B 113	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC p	1B 1B 114	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC q	1B 1B 115	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC r	1B 1B 116	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC s	1B 1B 117	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC t	1B 1B 118	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC u	1B 1B 119	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC v	1B 1B 120	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC w	1B 1B 121	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC x	1B 1B 122	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC y	1B 1B 123	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC z	1B 1B 124	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC A	1B 1B 125	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC B	1B 1B 126	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC C	1B 1B 127	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC D	1B 1B 128	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC E	1B 1B 129	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC F	1B 1B 130	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC G	1B 1B 131	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC H	1B 1B 132	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC I	1B 1B 133	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC J	1B 1B 134	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC K	1B 1B 135	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC L	1B 1B 136	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC M	1B 1B 137	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC N	1B 1B 138	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC O	1B 1B 139	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC P	1B 1B 140	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC Q	1B 1B 141	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC R	1B 1B 142	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC S	1B 1B 143	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC T	1B 1B 144	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC U	1B 1B 145	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC V	1B 1B 146	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC W	1B 1B 147	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC X	1B 1B 148	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC Y	1B 1B 149	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC Z	1B 1B 150	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC [1B 1B 151	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC \	1B 1B 152	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC]	1B 1B 153	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC ^	1B 1B 154	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC _	1B 1B 155	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC `	1B 1B 156	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC a	1B 1B 157	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC b	1B 1B 158	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC c	1B 1B 159	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC d	1B 1B 160	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC e	1B 1B 161	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC f	1B 1B 162	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC g	1B 1B 163	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC h	1B 1B 164	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC i	1B 1B 165	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC j	1B 1B 166	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC k	1B 1B 167	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC l	1B 1B 168	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC m	1B 1B 169	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC n	1B 1B 170	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC o	1B 1B 171	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC p	1B 1B 172	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC q	1B 1B 173	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC r	1B 1B 174	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC s	1B 1B 175	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC t	1B 1B 176	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC u	1B 1B 177	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC v	1B 1B 178	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC w	1B 1B 179	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC x	1B 1B 180	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC y	1B 1B 181	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC z	1B 1B 182	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC A	1B 1B 183	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC B	1B 1B 184	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC C	1B 1B 185	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC D	1B 1B 186	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC E	1B 1B 187	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC F	1B 1B 188	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC G	1B 1B 189	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC H	1B 1B 190	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC I	1B 1B 191	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC J	1B 1B 192	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC K	1B 1B 193	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC L	1B 1B 194	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC M	1B 1B 195	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC N	1B 1B 196	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC O	1B 1B 197	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC P	1B 1B 198	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9
ESC ESC Q	1B 1B 199	aa.. CR	-	TVI/TX	Byte-Folge empfangen	2.9
ESC ESC R	1B 1B 200	aa.. CR	-	TVI/TX	Byte-Folge senden	2.9

Befehl	Hex	Parameter	Antwort	Modus	Funktion	Handbuch-Seite
^G	07	-	-	TVI	Glocke	2.17
^H	08	-	-	TVI	Cursor nach links	2.16
^I	09	-	-	TVI	Cursor nach rechts	2.16
^J	0A	-	-	TVI	Zeilenvorschub	2.16
^K	0B	-	-	TVI	Cursor auf	2.16
^L	0C	-	-	TVI	Schirm initialisieren	2.17
^M	0D	-	-	TVI	Wagenrücklauf	2.16
^N	0E	-	-	TVI	Cursor ab	2.16
^O	0F	-	-	TVI	Schirm löschen	2.17
^P	10	-	-	TVI	Cursor Home	2.17
^Q	11	-	-	TVI	Neue Zeile	2.16
-	-	-	-	-	-	-
ESC \$	1B 24	-	-	TVI	Strichgraphik ein	

G R I P - A d r e s s b e l e g u n g

Der Speicherbereich ist folgendermaßen aufgeteilt:

0000h-3FFFh	EPROM
4000h-4800h	CPU-RAM
8000h-FFFFh	VIDEO-RAM

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.

Jeweils acht aufeinanderfolgende Bytes im Video-RAM bilden auf dem Schirm einen rechteckigen Block von 64 (8x8) Pixels. Die einzelnen Bytes bilden von oben nach unten die Pixelzeilen des Blocks. Ein Ausschnitt des Bildschirms, beginnend an einer Blockgrenze mit Byte N, ist dem RAM nach folgendem Schema zugeordnet:

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

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

Port	R/W	Bit	Name	Funktion
00h	--	--	/STB	Strobe-Signal für die Centronix-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	CC3	Ausgangs-Steuerleitung (CCON), auch zum Abschalten der Centronix-Schnittstelle (s.J2).

Port	R/W	Bit	Name	Funktion
11h	W	7	VOLO	Regelt in Kombination mit VOL1 die Lautstärke des Tongenerators nach folgender Tabelle: VOL1-0: 00 01 10 11 Ton : aus leise normal laut
12h	W	7	/RTS	Handshake-Signal zur V24/RS232-Schnittstelle 0: Bereit zum Datenempfang 1: Warten
13h	W	7	PAGE	Schaltet den Zugriff von CPU und VDC zwischen der unteren und oberen 32K-Page des Video-RAM's um. 0: Video-RAM Page 0 selektiert 1: Video-RAM Page 1 selektiert
14h	W	7	CC1	Ausgangs-Steuerleitung für Zusatzfunktionen
15h	W	7	CC2	Ausgangs-Steuerleitung für Zusatzfunktionen
16h	W	7	FLASH	Regelt den Zugriff auf das Video-RAM. 0: Zugriff von VDC 1: Zugriff von CPU (Bild dunkel)
17h	W	7	VOL1	Tongenerator-Lautstärke (s. VOLO)
20h	R/W	0-7	IDR	Zugriff auf die indirekten STI-Register
21h	R	0	/CTS	Handshake-Signal von der V24/RS232-Schnittstelle 0: Senden möglich 1: Warten
21h	R	1	DE	Position des Elektronenstrahls 0: Strahl im Dunkelfeld am Bildschirmrand 1: Strahl im Hellfeld
21h	R	2	CU	Cursor-Impuls; Cursorposition einstellbar über VDC-Register 14 und 15 0: Strahl außerhalb der Cursorposition 1: Strahl auf Cursorposition (ca. 500 ns)
21h	R	3	BUSY	Handshake-Signal vom Centronix-Interface 0: Senden möglich 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	EXIN	Takteingang für IBM-Tastatur

Port	R/W	Bit	Name	Funktion
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-Maskenregister 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 (Melodiegenerator-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

Port	R/W	Bit	Name	Funktion
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 VOLO (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 vom Centronix-Interface 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

G R I P - J u m p e r

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

1. EPROM-Typ

2732 (4 KByte): J1: Pos. 1-2
2764 (8 KByte): J1: Pos. 2-3 (default)
27128 (16 KByte): J1: Pos. 2-3 (default)

2. Datenausgänge der Centronix-Schnittstelle

Bei speziellen Anwendungen kann es sinnvoll sein, die Ausgänge DATA1-8 in den hochohmigen Zustand zu versetzen. Dafür ist J2 und das Steuerflag CC3 vorgesehen. Es lassen sich die folgenden Modi einstellen:

MODUS 1: Die Ausgänge DATA1-8 sind immer durchgeschaltet; CCON ist eine Ausgangsleitung, die von Flag CC3 gesteuert wird. Dieser Modus ist voreingestellt.

MODUS 2: Die Ausgänge DATA1-8 werden hochohmig, sobald CC3 auf "1" gesetzt wird. CCON verhält sich wie in MODUS 1.

MODUS 3: CCON ist ein Eingang, der die Ausgänge DATA1-8 abschaltet, wenn ein HIGH-Pegel angelegt wird. CC3 hat hier keine Funktion.

MODUS 1: J2: Pos. 1-2,3-4 (default)
MODUS 2: J2: Pos. 1-2,2-3
MODUS 3: 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. Über den ersten Jumper läßt sich die Art der Ansteuerung durch den Hauptrechner (Host) einstellen.

Pos.	Host-Kanal
offen:	Host = ECB-Bus
1-2:	Host = V24, 9600 Baud, 8 Bit (RTS/CTS benutzen!)
2-3:	Host = V24, 4800 Baud, 8 Bit
4-5:	Host = V24, 1200 Baud, 8 Bit
5-6:	Host = Tastatur (Lokal-Modus)

Im Lokal-Modus werden die Daten von der Tastatur als Befehle interpretiert; auf diese Weise kann man die Karte auch ohne angeschlossenen

GRIP - Jumper

Host-Rechner austesten.

Der zweite Jumper bestimmt die Art der Tastatur. Es lassen sich parallele und serielle ASCII-Tastaturen mit TTL- oder V24-Pegel und verschiedenen Baudraten und Datenlängen an GRIP betreiben. Die Polarität der Daten kann per Konfigurationsbefehl bestimmt werden.

Pos. Tastatur-Art

offen: Tastatur = parallel mit Strobe, nichtinvertiert
7-8: Tastatur = seriell, 1200 Baud, 8 Bit
8-9: Tastatur = seriell, 1200 Baud, 7 Bit
10-11: Tastatur = seriell, 600 Baud, 8 Bit
11-12: Tastatur = seriell, 600 Baud, 7 Bit

4. Zusatzkarte

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 am ECB-Bus (c-26) 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.

GRIP - Jumper

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

8. PAGE-Flag

Normalerweise ist nur die Hälfte des Video-RAM's auf dem Bildschirm sichtbar, die durch das PAGE-Flag adressiert wurde. Wenn J8 umgesteckt wird, kann das gesamte Video-RAM angezeigt werden. Dies ist beim Zeilensprung-Verfahren (interlaced) erforderlich, da der sichtbare Teil des Speichers dann größer als 32 KByte ist. Auf die Adressierung des Speichers von der CPU her hat J8 keinen Einfluß.

Bildgröße max. 32 KByte: J8: Pos. 1-2 (default)
Bildgröße max. 64 KByte: J8: Pos. 2-3 (erforderlich für GRIP-3)

9. CPU-Taktrate

2 MHz: J9: Pos. 2-3
4 MHz: J9: Pos. 1-2 (default)

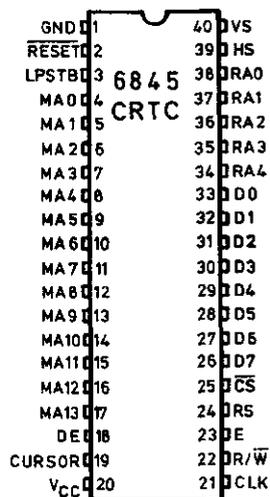
10. Pixel-Taktfrequenz

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.

P r o g r a m m i e r u n g d e s 6 8 4 5

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 Synchronisierung 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 bedeuten folgendes:



GND:	Masse
VCC:	+5V-Spannung
DO-D7:	Datenleitungen
E:	Chip-Select-Eingang
/CS:	Chip-Select-Eingang
R/W:	Lesen/Schreiben
RS:	Registerauswahl
/RESET:	Rücksetz-Eingang
MA0-13:	Block-Adressen
RA0-4:	Byte-Adressen
CLK:	Takteingang
LPSTB:	Lichtgriffeleingang
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 Rx einprogrammierte Wert. Die angegebenen Zeiten beziehen sich auf eine 1MHz-Frequenz am CLK-Eingang des 6845; alle internen Frequenzen sind davon abgeleitet.

Programmierung des 6845

R0: Zeilenlänge. Die Dauer einer Zeile inklusive Rücklaufperiode beträgt $(R0)+1$ 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 Mikrosekunden, 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 des horizontalen Sync-Impulses in Mikrosekunden. Bei einer eingeschriebenen "0" wird das horizontale Sync-Signal ganz unterdrückt. Die Dauer des vertikalen Sync's ist fest auf $16*((R0)+1)$ 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) gewählt werden.

R7: Bildlage. Die unteren 7 Bit von (R7) legen die Position des vertikalen Sync-Impulses in Einheiten von $((R0)+1)*((R9)+1)$ 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. 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.

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

Programmierung des 6845

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 die untersten Bits (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.

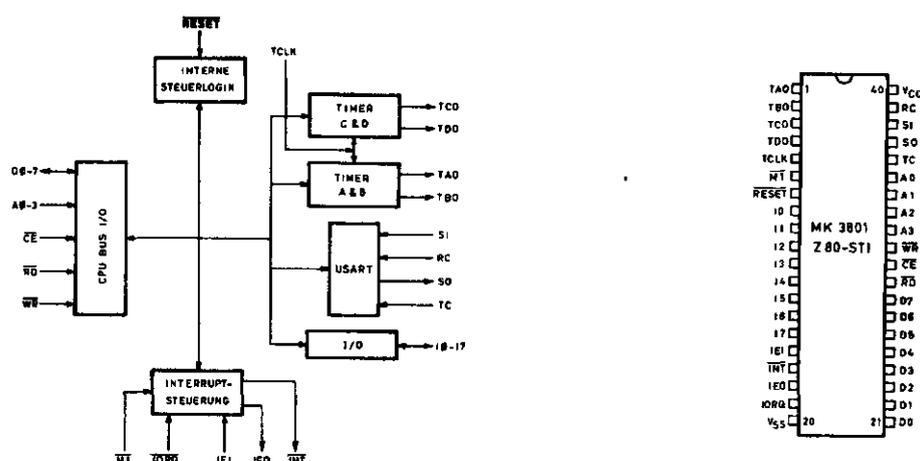
In alle Register, bis auf R16 und R17, lassen sich Daten einschreiben; gelesen werden kann jedoch nur aus den Registern R12-R17.

Lit.: Motorola 6845 CRTC Data Manual

P r o g r a m m i e r u n g d e s S T I

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 eingesetzt 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	TBR5	TDC2	TDC1	TDC0

P r o g r a m m i e r u n g d e s S T I

Adresse	Nummer	Name	Bit 7	6	5	4	3	2	1	0
21h	R1	GPIP	DP7	DP6	DP5	DP4	DP3	DP2	DP1	DPO
22h	R2	IPRB	IPP5	IPP4	IPTC	IPTD	IPP3	IPP2	IPPI	IPPO
23h	R3	IPRA	IPP7	IPP6	IPTA	IPRB	IPRE	IPXB	IPXE	IPTB
24h	R4	ISRB	ISP5	ISP4	ISTC	ISTD	ISP3	ISP2	ISPI	ISPO
25h	R5	ISRA	ISP7	ISP6	ISTA	ISRB	ISRE	ISXB	ISXE	ISTB
26h	R6	IMRB	IMP5	IMP4	IMTC	IMTD	IMP3	IMP2	IMP1	IMPO
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	WLO	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)

Programmierung des STI

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:

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

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

IP_{xx}: 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.

IS_{xx}: 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 GPIIP-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 TACO-2, TBCO-2, TCCO-2 und TDCO-2 nach folgender Tabelle eingestellt:

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

Programmierung des STI

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 (TxCO-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

$$f = T/(2 \cdot Z),$$

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 Melodiegenerator benutzt, Timer C und D liefern die Baudraten für Sender und Empfänger 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.

STI	STO	Betriebsart
0	0	Synchron
0	1	Asynchron, 1 Stopbit
1	0	Asynchron, 1 1/2 Stopbits
1	1	Asynchron, 2 Stopbits
WLI	WLO	Wortlänge für Sender und Empfänger
0	0	8 Bit
0	1	7 Bit
1	0	6 Bit
1	1	5 Bit
SOL	SOO	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

Programmierung des STI

und PEO bestimmt.

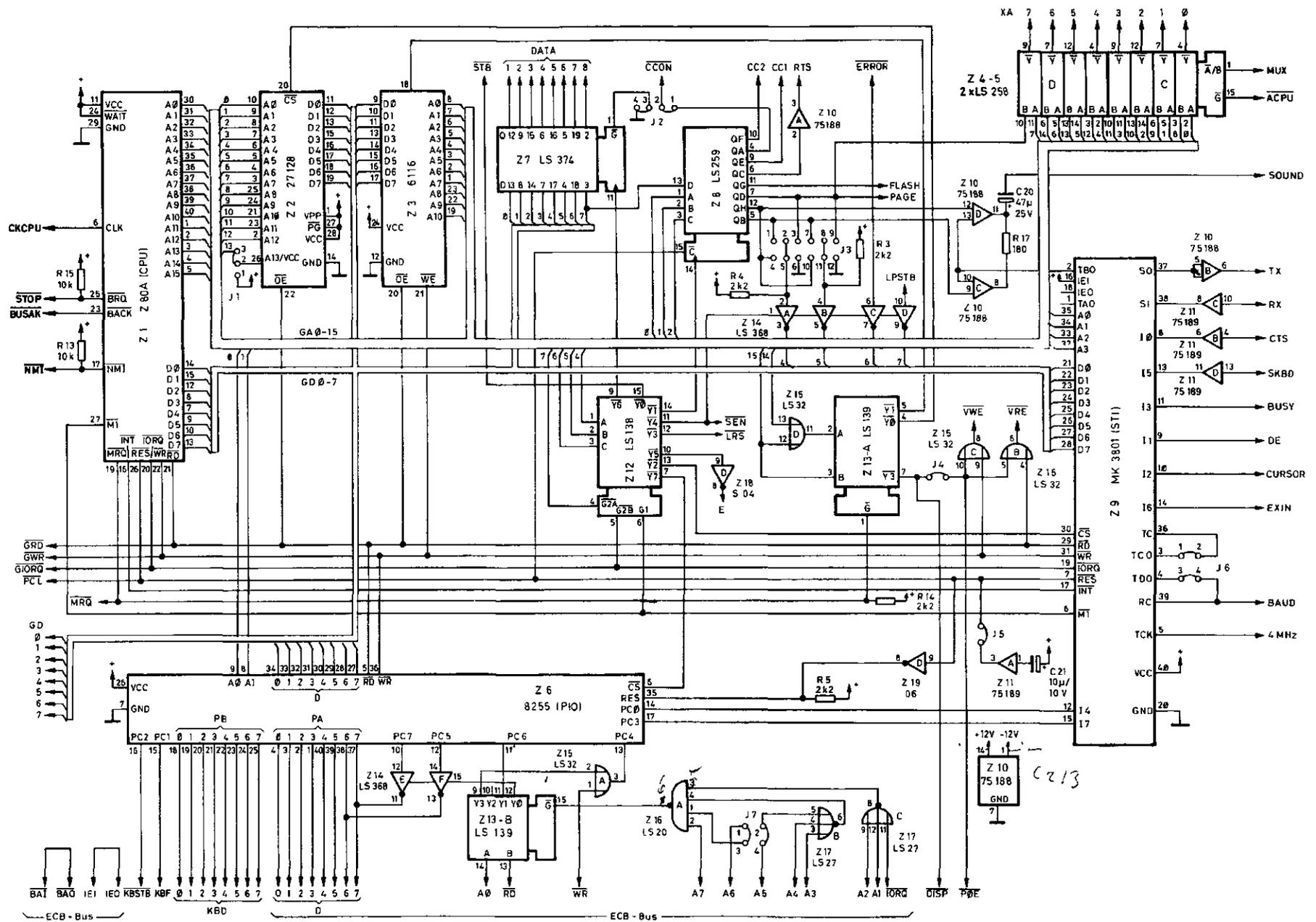
PEN PEO	Parität
0 0	Kein Paritätsbit
0 1	Kein Paritätsbit
1 0	Ungerade Parität
1 1	Gerade Parität

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)

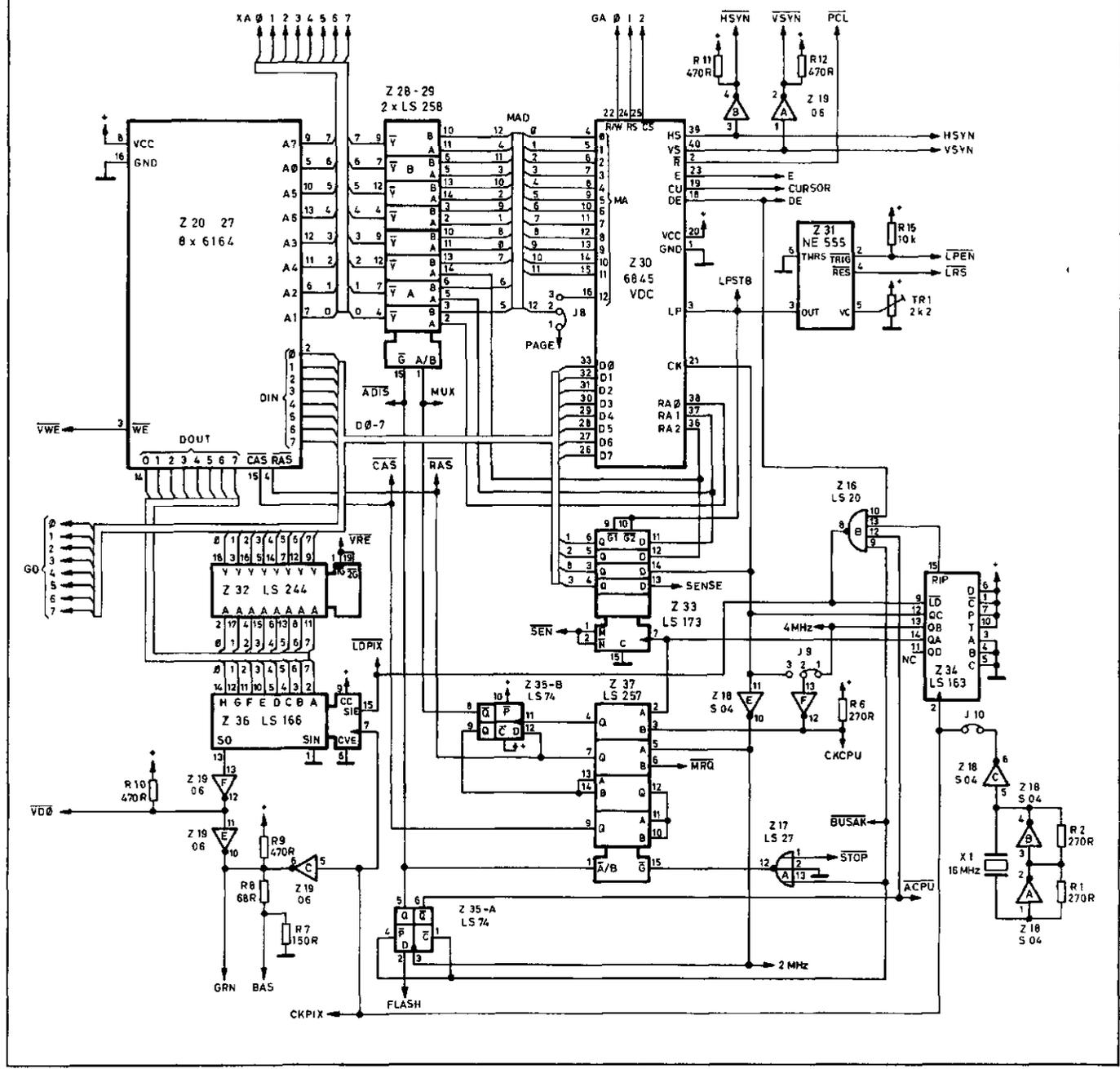
Im DMA-Handshake-Betrieb liegen die Signale RBF und TBE an den I/O-Leitungen 0 und 1. Auf GRIP ist DMA-Betrieb nicht vorgesehen, so daß das DMAE-Bit immer auf "0" programmiert werden muß. RBF springt auf "1", sobald ein empfangenes Datenbyte zum Lesen zur Verfügung steht. Geht TBE auf "1", kann ein Byte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

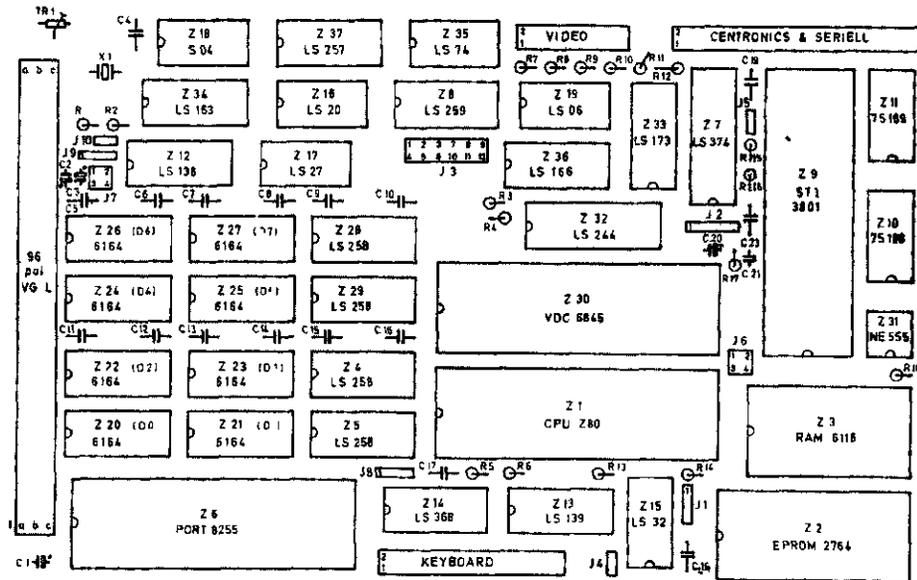
Literatur: Mostek, STI MK3801 Data Manual



Dieser Schaltungsteil zeigt den Prozessorbaustein und die Schnittstellen

Grafikteil der Schaltung: Video-Controller und Bildspeicher





GRIP-1 Stückliste

Bauteil	Typ	Funktion	Bauteil	Typ	Funktion
R8	68 Ohm	Widerstand	Z16	74LS20	2xNAND, je 4 Eing.
R7	150 Ohm		Z17	74LS27	3xNOR, je 3 Eing.
R17	180 Ohm		Z15	74LS32	4xOR, je 2 Eing.
R1, R2, R6	270 Ohm		Z35	74LS74	2xD-Flipflop
R9	470 Ohm	Bestimmt den BAS-Signalpegel	Z12	74LS138	Dekoder 8 aus 3
R10-R12	470 Ohm		Z13	74LS139	2xDekoder 4 aus 2
R3-R5, R14	2.2 kOhm		Z34	74LS163	Zähler 4 Bit
R16, R18	2.2 kOhm		Z36	74LS166	Schieberegister
R13, R15	10 kOhm		Z33	74LS173	Latch 4 Bit
TR1	2.2 kOhm	Lichtgriffel-Trimmer	Z32	74LS244	Buffer Tristate
			Z37	74LS257	MUX Tristate
			Z4, Z5,		
			Z28, Z29	74LS258/259	MUX Tristate, inv.
			Z8	74LS259	DEMUX Latch
			Z14	74LS368	Buffer Tristate
			Z7	74LS374	Latch 8 Bit
			Z11	1489/75189	V24-Empfänger
			Z10	1488/75188	V24-Treiber
			Z31	555	Timer
			X1	15 MHz	Quarz
			1x	8-pol.	IC-Sockel
			8x	14-pol.	"
			19x	16-pol.	"
			2x	20-pol.	"
			1x	24-pol.	"
			1x	28-pol.	"
			4x	40-pol.	"
			N1	96pol.	VG-Leiste
			N2	16pol.	Pfostenleiste
			N3	34pol.	"
			N4	26pol.	"
Z1	28400A	Z80A-CPU			
Z6	8255	I/O-Port			
Z9	MK3801-4	Z80A-STI			
Z30	6845	VDC			
Z2	27128	16KByte-EPROM			
Z3	6116	2KByte-RAM			
Z20-Z27	6665-3	64KBit-DRAM			
Z18	74S04	6xInverter			
Z19	7406/05	6xInverter m. o.K.			

Steckerbelegung

N1: ECB-Bus-Stecker

a	b	c	Leitung	Typ	Funktion
1:	+5V	+5V	+5V;	I, U	Betriebsspannung
2:	D5	/POE	+12V, -12V:	I, U	V24-Spannungen nicht benutzt
3:	D6	/GRD	-5V:	I, U	gemeinsame Masse
4:	D3	/GWR	GND:	I, U	Externer Datenbus
5:	D4	GDO	DO-D7:	IO, TS	Externer Adressbus
6:	A2	GD1	AO-A7:	I	Extern Lesen/Schreiben
7:	A4	XAO	/RD, /WR:	I	Externe Ein/Ausgabe
8:	A5	XA6	/IORQ:	I	Vorrang-Interrupt
9:	A6	GD2	/NMI:	I	Karte rücksetzen
10:	xxx	GD3	/PCL:	I/0, OD	
11:	xxx	XA1	IEI, IEO:	I, O	INT-Kette
12:	xxx	XA2	/BAI, /BAO:	I, O	DMA-Kette
13:	+12V	/DIS	GDO-7:	IO, TS	Graphik-Datenbus
14:	-12V	GD4	XAO-7:	IO, TS	Graphik-Adressbus
15:	-5V	GD5	/DISP:	0	8000h-FFFFh Select
16:	-	XA5	/POE:	I/0	Video-RAM Select
17:	xxx	XA7	/GRD, /GWR:	IO, TS	Intern lesen/schreiben
18:	xxx	GD6	/GIORQ:	IO, TS	Interne Ein/Ausgabe
19:	-	GD7	/STOP:	I	Interner DMA-Zugriff
20:	xxx	/CAS	/ADTS:	0	VDC-Zugriff
21:	xxx	XA3	/RAS, /CAS:	IO, TS	RAM-Steuersignale
22:	-	XA4	CKPIX:	I/0	Bildpunktakt
23:	/BAI	/STOP	/LDPIX:	0	Ladetakt
24:	xxx	/GIORQ	/VDO:	0, OD	Video-Signal
25:	/BAO	/RAS	VSYN, HSYN:	0	Sync-Signale
26:	xxx	/VDO			
27:	/IORQ	VSYN			
28:	xxx	HSYN			
29:	xxx	/ADIS			
30:	xxx	/LDPIX			
31:	xxx	CKPIX			
32:	GND	GND			

xxx = reserviert für spätere Erweiterungen
 - = 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
GND:	0, U	gemeinsame Masse
BAS:	0, OD	Video-Signal
GREEN:	0, OD	Video-Signal TTL
VSYN, HSYN:	0	pos. Sync-Signale
/VSYN, /HSYN:	0, OD	neg. Sync-Signale
CKPIX:	I/O	Bildpunktakt (s.J10)
SOUND:	0, 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
+5V, -5V, +12V, -12V:	0, U	Versorgungsspannungen für externe Geräte
GND:	0, U	gemeinsame Masse
DATA1-8:	0, TS	Centronix-Daten
/STB:	0	Übergabe-Impuls
BUSY:	IO, TS	Nicht empfangsbereit
/INIT:	0, OD	Initialisieren
/ERROR:	I	Fehler
CCON:	I/O	Steuerleitung (s.J2)
TX:	0, V24	V24-Datenausgang
RX:	I, ST	V24-Dateneingang
RTS:	0, V24	Handshake-Ausgang
CTS:	I, ST	Handshake-Eingang
BAUD:	I/O	16facher Baudraten-Takt (s.J6)
KB1-8:	IO, TS	Tastatur-Daten
/KBSTB:	IO, TS	Übergabe-Impuls
KBF:	IO, TS	Nicht empfangsbereit
SKBD:	I, ST	Serieller Tastatur-Eingang
CC1-2:	0	Ausgangsleitungen
SENSE:	I	Eingangsleitung
EXIN:	IO, TS	Interruptleitung
2 MHz:	0	Taktsignal
SOUND:	0, A	NF-Ausgang
/LPEN:	I, A	Lichtgriffel-Impuls

N4: Schnittstellen

Leitung	Typ	Funktion
+5V	+5V	
DATA1	DATA2	
DATA3	DATA4	
DATA5	DATA6	
DATA7	DATA8	
/STB	BUSY	
/INIT	/ERROR	
CC1	CC2	
GND	GND	
19-20	GND	SOUND
21-22	2 MHz	SENSE
23-24	/LPEN	-5V
25-26	+12V	-12V
SKBD	BAUD	
27-28	TX	RX
29-30	TX	CTS
31-32	RTS	CTS
33-34	GND	GND
+5V	+5V	
KB1	KB2	
KB3	KB4	
KB5	KB6	
KB7	KB8	
/KBSTB	KBF	
EXIN	SENSE	
CC1	CCON	
GND	GND	
17-18	GND	SOUND
+12V	-5V	
2 MHz	2 MHz	SENSE
/LPEN	-5V	
GND	GND	