

```

*****
*****
**                                     **
**                                     **
**           Handbuch  TERM  1         **
**                                     **
**   Terminal und Graphik-Subsystem   **
**                                     **
**           Ausgabe 2       5/86      **
**                                     **
**                                     **
*****
*****

```

(c) by GES GmbH, 1983

Jegliche Vervielfältigung, auch auszugsweise, nur nach schriftlicher Genehmigung der GES GmbH gestattet.

Alle Rechte vorbehalten.

## Vorwort

-----

Dieses Handbuch ist für TERMI - Benutzer, die das System als Platine, Bausatz oder Fertiggerät gekauft haben, sowie für Anwender des Graphik-Subsystemes GSS gleichermaßen gedacht.

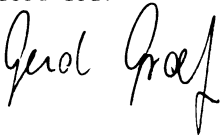
Die erstere Gruppe sollte dieses Handbuch Schritt für Schritt durcharbeiten.

Ein Käufer des Graphik-Sub-Systemes GSS möchte vermutlich so schnell wie möglich etwas sehen - verständlich, dem Verfasser dieses Handbuches geht es immer genau so. Hier empfehlen wir, trotzdem das Handbuch in der vorliegenden Reihenfolge zumindest zu überblättern, um dann in letztem Kapitel "Anwendungsbeispiele" zusammen mit dem vorletzten Kapitel "Befehlsbeschreibung" die ersten Erfolgserlebnisse zu genießen.

Dem Wunsch vieler Anwender folgend, haben wir die Original-Unterlagen des Thomson-Graphik-Display-Prozessors GDP 9366 mit freundlicher Genehmigung des Herstellers mit abgedruckt. Das Graphik-Sub-System ist vom Werk aus auf 4800 bd eingestellt.

Kempton, im August 1983

Gerd Graf



## Was ist TERM1?

-----

Mit dem Terminal TERM1 , das ursprünglich nur für den mc-CP/M-Computer geplant war, hat Rolf-Dieter KLEIN, der Entwickler dieses Gerätes, ein Produkt geschaffen, das seinesgleichen in der Mikrocomputerwelt sucht.

Wir sind es nicht gewohnt, solch "starke Worte" in einer technischen Beschreibung zu verwenden, aber bei diesem Gerät können wir nicht anders.

## TERM1 ist:

- \* Ein komplettes Terminal mit Editierfunktionen, wie Scrolling, Zeilen oder Zeichen einfügen oder löschen etc; Darstellung von 24 Zeilen mit je 80 Zeichen, programmierbarer Charaktersatz (ASCII oder deutsche Umlaute und ß)
- \* Ein vollständiges Graphik-Interface mit hochauflösender Graphik ( 512 x 256 Punkte ), das aber...
- \* Vektorgraphik kann, durch den superschnellen Graphik-Display Prozessor GDP 936x von Thomson-CSF, der bis zu 1,2 Mio Bildpunkte pro Sekunde verarbeitet.

## TERM1 ist aber noch intelligenter durch :

- \* Eine eigene CPU ( Z 80A ), verbunden mit 8 KByte EPROM mit Graphik-Routinen; dadurch kann TERM1 z.B.:
- + Linien ziehen ohne Begrenzung über den gesamten Arbeitsbereich
- + Neben Linien auch Rechtecke, Polygone und ...
- + Kreise oder Ellipsen zeichnen, natürlich auch Teile davon.
- \* TERM1 verfügt über einen Bildwiederholtspeicher von 64 KByte auf der Platine und kann damit:
- + Vier Bildschirmseiten (Pages) speichern
- + Eine Seite anzeigen, während gleichzeitig in eine andere Seite geschrieben wird
- + Zwei oder alle vier Seiten mit wählbarer Geschwindigkeit umschalten - damit lassen sich bewegte Bilder erzeugen, Fadenkreuze einblenden oder Bildteile übereinander schieben

- \* TERM1 ist über eine serielle V24-Leitung anzusteuern und damit an jeden Mikroprozessor mit V24-Schnittstelle anzuschliessen!
- \* TERM1 verfügt aber über eine weitere V24-Schnittstelle - dort kann eine Tastatur angeschlossen werden, um ein komplettes Terminal zu erzeugen.
- \* TERM1 ist auf einer Europakarte
- \* TERM1 ist ECB-Bus-kompatibel, benötigt jedoch vom Bus nur drei Versorgungsspannungen (+5, +/-12V) und Masse
- \* TERM1 ist selbstverständlich für Farbe vorbereitet!

## Technische Daten:

-----

**Master-CPU:** Z80A  
**Slave-CPU** GDP 9366

**Systemspeicher:** 8 KByte EPROM 2764  
**System-RAM** 4 KByte RAM 6116, vorbereitet für 16K

**Bildwiederholtspeicher:** 64 KByte für 4 Pages, dyn.RAM 4164

**Anschluss:** 2 V24-Schnittstellen, Baudraten frei zwischen 50 und 9.600 bd wählbar (optional 14400 und 28800 bd)  
 Eine Schnittstelle zum Host-Rechner, eine für Tastaturanschluss vorgesehen

**Befehle Terminal-Mode:** CURSOR Wähle Cursor  
 (Auszug) INSERT CHAR. Füge Zeichen ein  
 DELETE CHAR. Lösche Zeichen  
 INSERT LINE Füge Zeile ein  
 DELETE LINE Lösche Zeile  
 ERASE TO EOL. Lösche bis Zeilenende  
 ERASE TO EOP. Lösche Bis Seitenende  
 CLEAR Lösche alles  
 POS CURSOR Fahre Cursor nach oben, unten, rechts, links, home  
 SWITCH Umschalten zur Graphik

**Befehle Graphik-Mode:** MOVE TO Bewege  
 (Auszug) DRAW TO Zeichne zu  
 DRAW RELATIV Zeichne weiter  
 RECHTECK Zeichne Rechteck  
 KREIS Zeichne Kreis, Ellipse, auch Teile  
 FADENKREUZ Zeichne Fadenkreuz  
 PAGE Definiere Seite  
 SWITCH PAGE Schalte Seite um  
 DIREKT Übergebe Befehle an GDP  
 TEXT Schreibe Text  
 SWITCH TERM Umschalten Terminal-Mode

T4014 Befehle im TEKTONIX T4014-Mode!

Zum Handbuch  
-----

Das vorliegende Handbuch soll Ihnen als Entscheidungshilfe dienen, falls Sie es vorerst allein bestellt haben. Sie bekommen den Kaufpreis des Handbuches ja bei einer späteren Bestellung gutgeschrieben - vergessen Sie bitte nicht, dann die Rechnungsnummer anzugeben!

Falls Sie eine Platine oder einen Bausatz mitbestellt haben, soll Ihnen das Handbuch beim Aufbau oder, falls nötig, bei der Fehlersuche helfen.

Zum Betrieb des Gerätes TERM1 dient Ihnen ebenfalls dieses Handbuch - Sie werden dann sicher das Kapitel "Programmieren des Systems" benötigen.

Beachten Sie bitte, dass das Handbuch nur zu Ihrem persönlichen Gebrauch bestimmt ist. Kommerzielle Anwendung der veröffentlichten Schaltung, der Layouts und der Systemsoftware sowie Vervielfältigung, Veröffentlichung usw., auch auszugsweise, sind nur mit unserer ausdrücklichen schriftlichen Zustimmung erlaubt. Dies jedoch nur zu Ihrer Information.

Wir haben uns bemüht, das vorliegende Handbuch fehlerfrei zu halten, und sind uns gleichzeitig sicher, dass uns dies nicht gelungen ist. Wir bitten deshalb um Ihre Mitarbeit - sollten Sie Fehler entdecken, teilen Sie uns diese doch bitte gleich mit - am Besten auf einer Kopie der jeweiligen Seite. Vielen Dank!

#### Prüfen auf Vollständigkeit

Packen Sie zunächst Ihre Sendung aus und prüfen Sie an Hand der folgenden Stücklisten, ob Sie auch alle Teile erhalten haben.

-----

**ACHTUNG !** MOS-Bauteile sind sehr empfindlich gegen statische Aufladungen! Berühren Sie die Anschlussbeine der RAMs und der IC's, die getrennt verpackt sind, möglichst nicht oder nur dann, wenn Sie sich vorher entladen haben (Erde berühren, z.B. Schutzkontakt)

-----

/ -6-



## Stückliste für TERM1

- 
- 1 Original GES-Platine mit Lötstoplack  
1 Handbuch

ACHTUNG ! Leider sind einige schwarze Schafe auf die Idee gekommen, unsere Platinen nachzumachen und, meist billiger, zu verkaufen. Seien Sie bitte vorsichtig! Diese Platinen sind natürlich nicht von der Originalvorlage (2:1), die wir erstellt haben, gefertigt, sondern aus der veröffentlichten 1:1-Vorlage abgekupfert. Sie sind damit minderer Qualität und können Probleme bereiten!

Kaufen Sie in Ihrem Interesse Platinen NUR BEI UNS!

HINWEIS: Hilfe bei der Fehlersuche können wir nur dann geben, falls Sie die Ware bei uns oder bei einem autorisierten Distributor bezogen haben! Der günstige Pauschalkosten satz für Reparaturen gilt natürlich auch nur für von uns bezogene Ware.

Stückliste TERM1 66

Ausgabe 2  
28.01.1986

- 1 Original GES-Platine mit Lötstoplack TERM1 r1
- 1 Handbuch 3. Auflage
- |   |            |           |                                  |
|---|------------|-----------|----------------------------------|
| 1 | 7404       | J23       | 6 Inverter                       |
| 1 | 74LS00     | J27       | 4*2 NAND                         |
| 1 | 74LS04     | J4        | 6 Inverter                       |
| 1 | 75LS05     | J24       | 6 Inverter, Open Collector       |
| 1 | 74LS32     | J12       | 4*2 OR                           |
| 1 | 74LS74     | J11       | 2 D Flip-Flop                    |
| 1 | 74LS139    | J2        | 2*2-Bit Binärdecoder             |
| 1 | 74LS153    | J14       | 2* 4 zu 1 Datenselektoren        |
| 1 | 74LS163    | J26       | 4 Bit Zähler                     |
| 1 | 74LS166    | J25       | 8 Bit Schieberegister            |
| 1 | Z80A       | J1        | CPU                              |
| 1 | 2764       | J5        | Eprom                            |
| 2 | Z80 STI    | J6, J8    | serieller Schnittstellenbaustein |
| 2 | 6116       | J7, J9    | statisches RAM, 8 * 2kBit        |
| 8 | 4164       | J15 - J22 | dynamisches RAM, 1 * 64kBit      |
| 1 | AM25LS2538 | J13       | 3 zu 8 Decoder                   |
| 1 | GDP 9366   | J10       | Grafik Prozessor                 |
| 1 | 75 189     | J28       | V24 Empfänger, auch MC1489       |
| 1 | 75 188     | J29       | V24 Sender, auch MC 1488         |
| 1 | 55         |           | J3 Timer                         |

/ -7-



2	10 nF	C4, C7	Keramik Kondensatoren
2	100nF	C2, C6	Keramik Kondensatoren
1	1uF/16V	C3	Tantal ELKO
3	10uF/16V	C1, C5, C8	Tantal ELKO
1	75 Ohm	R14	Widerstände
1	150 Ohm	R10	"
2	330 Ohm	R1, R7	"
1	470 Ohm	R8	"
8	1 kOhm	R2, R4 - R6, R9, R11 - R13	"
1	100 kOhm	R3	"
1	8*3,3 kOhm	RN1	Netzwerkwidestand
1	SO 8		IC-Fassung
8	SO 14		"
12	SO 16		"
1	SO 20		"
3	SO 28		"
4	SO 40		"
1	ST1		VG64-Stiftleiste, gewinkelt
1	ST2		1*6polige Stiftleiste gewinkelt
1	ST3		1*10polige Stiftleiste gewinkelt
1	BSX 20	TR1	Transistor
1	7,3728 MHz	Q1	Quarz
1	14 MHz	Q2	Quarz Grundwelle

## Stückliste in Bauteilfolge

J1	Z80 A	CPU 8 Bit
J2	74LS139	Zweifach Demultiplexer
J3	555	Timer
J4	74LS04	6-fach Inverter
J5	2764	EPROM 2764, 8KByte, programmiert
J6	Z80-ST1	Serial/Parallel-Interface
J7	6116	RAM, statisch, 2 KByte (auch 5117)
J8	Z80-ST1	
J9	6116	
J10	GDP 936x	Graphik Display Prozessor 9365 oder 9366
J11	74LS74	2 D-FF mit Preset und Clear
J12	74LS32	4-fach OR, zwei Eingänge
J13	AM25LS2538	Decoderbaustein
J14	74LS153	Zweifach 4 zu 1 Multiplexer
J15	.	
J16	.	
J17	4164	dyn. RAM 64 KBit, min 200 ns

/ -8-





J18	.	(auch 8264 o.ä.)
J19	.	
J20	.	
J21	.	
J22	.	
J23	7404	Kein LS-Typ!
J24	74LS05	Sechsfach-Inverter, open Collector
J25	74LS166	8 Bit Schieberegister mit Paralleleingang
J26	74LS163	Synchroner programmierbarer 4 Bit Zähler
J27	74LS00	4-fach NAND
J28	75189	V24-Sender
J29	75188	V24-Empfänger

R1	330	Org/Org/Brn
R2	1k	Brn/Schw/Rot
R3	100K	Brn/Schw/Gelb
R4	1k	
R5	1k	
R6	1k	
R7	330	Gelb/Vio/Brn
R8	470	
R9	1K	
R10	150	Brn/Grn/Brn
R11	1k	
R12	1K	
R13	1K	
R14	75 Ohm	vio/Grn/Schw
RN1	8 x 3,3 K	Widerstandsnetzwerk

alle Widerstände 1/4 W

C1	10 uF	Elko Tantal, blau ; u= Mikrofarad
C2	100 nF	; n= Nanofarad
C3	1 uF	Elko Tantal
C4	10 nF	
C5	10 uF	
C6	100 nF	
C7	10 nF	
C8	10 uF	uF=Mikrofarad, nF=Nanofarad

Bitte überprüfen Sie den Bausatz und identifizieren Sie alle Teile eindeutig. Die TTL-ICs lassen Sie bis zum Einsatz im Styropor stecken - die MOS-ICs sollten auch in Ihren Umhüllungen bleiben.

Sollte einmal ein Bauteil fehlen, so teilen Sie uns dies bitte mit - wir sind auch nur Menschen

Einige Widerstands- oder Kondensatorwerte können leicht von der Stückliste abweichen. Dies gilt besonders für die Elkos - hier sind Abweichungen um 100 % erlaubt.

Falls Sie nur eine Platine gekauft haben und Bauteile aus anderen Quellen verwenden - bitte verwenden Sie nur Qualitätsbauteile! Wir verwenden nur ICs von Texas Instruments, Präzisionssockel vom AMP usw. Das einwandfreie Funktionieren der Schaltung ist nur beim Einsatz von Qualitätsbauelementen gewährleistet.

Sollten Sie ein fehlendes, falsches oder vertauschtes Bauteil entdeckt haben, prüfen Sie nochmals genau die Verpackung.. Die Bausätze werden vor dem Versand geprüft. Sollte wirklich ein Bauteil fehlen, geben Sie uns bitte Bescheid.

Aufbau des TERM1

**Achtung:** wenn Sie für Ihre TERM eine Farberweiterung einsetzen wollen, darf der IC-Sockel für J23 u. J25 nicht bestückt werden. Es müssen die beim COLOR-Bausatz mitgelieferten MODU 2 Buchsenleisten eingesetzt werden.

Benötigtes Werkzeug:

LötKolben mit temperaturgeregelter Spitze  
Lötzinn, säurefrei, mit Kolophonium-Seele  
Pinzette  
Seitenschneider

Die Platine ist sehr eng bestückt. Bitte bauen Sie nur dann selber auf, wenn Sie über ausreichende Erfahrung im Aufbau von eng bestückten Platinen verfügen!

Ordnen Sie die Bauteile nach der Stückliste.

Prüfen Sie zunächst die unbestückte Platine. Achten Sie vor allem auf Verbindungen zwischen den Leiterbahnen, die durch Ätzfehler entstehen können. Prüfen Sie besonders sorgfältig die Bestückungsseite.

Legen Sie die Platine so vor sich, dass Sie die Beschriftungen (Z80, ST1A, ST1B usw) lesen können. Die Steckerleiste muss links liegen.

Die Lötseite der Platine ist mit "löts" bezeichnet. Sie muss unten sein.

Legen Sie die Platine mit der Lötseite auf ein Stück Styropor und bestücken Sie zunächst alle DIL-Sockel. Achten Sie bitte auf die Lage der Aussparungen - sie müssen alle nach links oder nach oben zeigen

Stecken Sie zunächst alle Sockel ein, ohne sie zu verlöten - nur so können Sie sicher sein, dass nicht ein 14-poliger Sockel am Platz eines 16-poligen steckt.

**HINWEIS:** Für die RAMs 6116 haben wir 28-polige Sockel vorgesehen, obwohl die Bausteine 24-polig sind. Grund: Für spätere Erweiterungen können hier 8 KByte RAMs 8464 eingesetzt werden. Die ICs werden bündig nach unten eingesteckt, oben bleiben also derzeit 4 Steckerpins frei.

Legen Sie ein Stück Karton über die Sockel, drehen Sie die Platine um und verlöten Sie zunächst von jedem Sockel zwei (gegenüberliegende) Beinchen. Drehen Sie die Platine weder um und prüfen Sie, ob alle Sockel gut anliegen. Hat sich beim Einsetzen ein Beinchen abgespreizt?

Ausgabe 2 / 05/86

/ -11-

Löten Sie nun alle Beinchen an.

Bestücken Sie nun die Steckerleiste und die passiven Bauteile.

Die Baugruppe belegt am ECB-Bus lediglich die Versorgungsspannungen +5V, + / - 12V und Masse. Sollten Sie einen anderen Bus haben, müssen Sie die Spannungen entsprechend umlegen. Sie können die Baugruppe auch OHNE Bus anschliessen - dazu sind diese Pins auch nach links unten geführt und dort gekennzeichnet.

Setzen Sie bitte noch keine ICs ein!!

Stecken Sie die Platine nun in Ihren Rechner.

#### Inbetriebnahme

Schalten Sie bei unbestückter Platine Ihren Rechner ein. Folgende Fehler können hier auftreten:

- Spannung bricht zusammen  
Vermutlich Kurzschluss zwischen zwei Versorgungsbahnen.

- Spannung bricht nach einiger Zeit zusammen  
Elko falsch eingelötet.

-Rechner geht nicht mehr  
Signalleitungen werden nach Masse oder + gezogen - falscher Anschluss am Bus

Messen Sie nun an einigen IC-Sockeln, besonders an den 40-poligen Sockeln alle Spannungen. Der GDP 9366 ist ein sehr teurer Baustein!

Bestücken Sie nun Ihre Platine mit den ICs. Achten Sie auf die richtige Lage aller ICs. Falls die Baugruppe mit dem Stecker nach links vor Ihnen liegt, sollten alle Beschriftungen von links nach rechts oder von oben nach unten lesbar sein.

#### Anschluss des Monitors

Verbinden Sie das TERM1 nun mit einem Video-Monitor und Ihrem Rechner. Dazu dienen die Steckerleisten an der rechten Seite der Platine, die wie folgt belegt sind:

Belegung der oberen Leiste:(Video):

-----	+5V	
-----	V*	Verikal Blank, Low Aktiv
-----	H*	Horizontal Sync
-----	D*	Video-Signal, TTL
-----	BAS	BAS-Signal
-----	Masse	

Einen SW-Video-Monitor (z.B. Sanyo) schliessen Sie einfach an BAS und Masse an. Verwenden Sie hier ein abgeschirmtes Kabel.

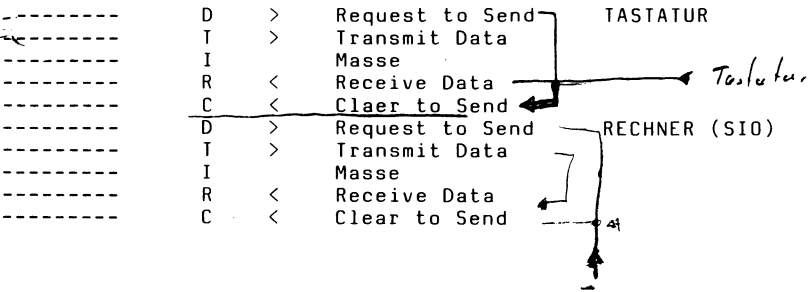
Sie müssen nun bereits einen blinkenden Cursor an der linken oberen Ecke des Bildschirmes sehen.

Anschluss an den Computer

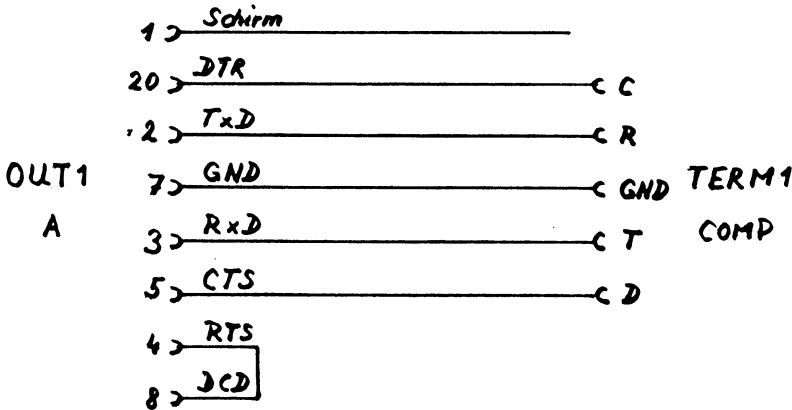
Der Computer, z.B. der mc-CP/M-Computer und die Tastatur werden an der unteren Steckerleiste angeschlossen.

Achtung Falle: Ist TERM1 in einen Einschub eingesteckt, ist oben und unte vertauscht! Achten Sie bitte hier auf die Beschriftung auf der Leiterplatte!

Belegung der unteren Leiste (V24):  
(Der einzelne Buchstabe ist auf der Leiterplatte beschriftet!)



Das folgende BILD zeigt den Anschluss an den mc-CP/M-Computer:



**Anschluss an den mc/CP/M-Computer**

Einstellen der Baudrate.

TERM ist, ohne dass Brücken gelegt werden müssen, auf 9600 bd eingestellt. Die Baudrateneinstellung erfolgt über J2(Bestückungsplan), wobei die Jumper von unten nach oben mit 7 bis 2 bezeichnet werden müssen. Die nächst J4 liegende Brücke ist also J7, die nächst J2 liegende Brücke J2. J2 ud 3 sind allerdings keine Brücken, d.h. sie werden zur Baudrateneinstellung nicht benötigt.

Bit	7	6	5	4	Baudrate
	0	0	0	0	28200
	0	0	0	1	14400
	1	1	1	1	9600
	1	1	1	0	4800
	1	1	0	1	2400
	1	1	0	0	1200
	1	0	1	1	600
	1	0	1	0	300
	1	0	0	1	110
	1	0	0	0	75
	0	1	0	0	50

"1" bedeutet "keine Verbindung", "0" eingelegte Brücke



Eine von 9600 bd abweichende Baudrate kann nur dann eingestellt werden, wenn **beide** STIs bestückt sind

Zum ersten Test können Sie, falls Ihr Computer bereits über ein Terminal verfügt, TERM1 wie einen Drucker anschliessen.

Vergessen Sie beim mc-CP/M-Rechner nicht "AL=L" nach dem RESET zu geben, und "drucken" Sie einige Zeilen.

Sie müssen am Bildschirm erscheinen.

Zeigt der Bildschirm nur wirre Zeichen an, so ist die Baudrateneinstellung zu überprüfen. Kommen gar keine Zeichen an, so ist die Spannung an CTS zu prüfen, sie muss auf +12V liegen.

Verschwanden Zeichen bei ausgabe von längeren Texten (mit Scrollen), so funktioniert das Handshake nicht. Am Computer ist nun CTS zu testen, die die Ausgabe anhalten muss, wenn sie auf -12V liegt.

Der Puffer von TERM1 beträgt 1024 Zeichen; die Ausgabe wird kurz vor dem Überlauf gesperrt.

Ihr TERM1 ist nun betriebsbereit.

Wird das TERM1 als Sichtgerät verwendet, so muss das CTS-Signal am mc-CP/M-Computer dafür programmiert sein. In der SIO-Tabelle muss der Status entsprechend geändert werden. Diese Änderung ist ab Monitor V 3.4 beinhaltet - bei älteren Monitoren muss in der SIO-Daten-Tabelle der Wert C1 auf E1Hex geändert werden.

## Beschreibung des TERM1

-----

TERM1 wird an Hand des Schaltplanes beschrieben. Bezeichnungen sind z.B. J2/14: dies bedeutet IC J2, Pin Nummer 14.

## Überblick

-----

TERM1 besteht aus zwei logischen Teilen:

- Dem Computerteil mit Z80, EPROM, RAM und seriellen Schnittstellen
- Dem Graphikteil mit GDP, Bildwiederholpeicher, Page-Logik und Mischer

## Der Computerteil

-----

J4 erzeugt mit Q1 einen Takt von 7.3728 MHz, der durch J11/11 durch zwei geteilt über J4/12 der CPU zugeführt wird.

Dieser Takt dient auch den Timern in den beiden STIs (Serial Timer Interrupt Controller) J6 und J8 als Arbeitstakt - dadurch lässt sich eine maximale Baudrate der V24-Schnittstelle von 24,8 Kilobaud erzeugen.

Der Timer J3 erzeugt einen kurzen Rücksetzimpuls beim Einschalten der Versorgungsspannung über R3 und C3. Das Rücksetzsignal wird an die CPU sowie an die STIs geführt.

IORQ\* und MREQ\* (\* bedeutet low-aktiv) dienen als Enable (Freigabe) der beiden Adressdecoder, die in J2 vereint sind. IORQ\* wird weiter an die STIs geführt.

Der linke der beiden Adressdecoder decodiert die Speicheradressen, der rechte die IO's. Die Speicheradressen A13 und A14 werden an J2/14,13 geführt und schalten einen der vier Ausgänge J12/12,11,10,9 auf Low. Drei der Ausgänge schalten die CS\*-Eingänge des EPROMs J5 und der RAMs J7 und J9.

Da das Adressbit A15 nicht mit verwendet wird, ist diese Decodierung nicht vollständig, d.h. das EPROM wird bei Adressen 0-1FFF, 8000-9FFF usw. aktiviert. Da diese Adressen aber vom Programm nicht verwendet werden und die Platine nicht erweitert werden kann (Platz), bedeutet dies keine Einschränkung.

Über J2/4,5,6,7 werden die IO-Bausteine adressiert. Es ergeben sich die im Schaltplan eingetragenen Adressen. Auch hier ist die Decodierung nicht eindeutig.



Über den Jumper J1 lassen sich (später) auch die Rams mit der doppelten Speicherkapazität einsetzen. Derzeit wird dieser RAM-Speicher noch nicht benötigt.

Die STIs J8 und J6 dienen zum Verkehr mit dem Rechner und der Tastatur. Die internen Timer erzeugen die Baudrate der V24-Schnittstellen; sie ist standardmässig auf 9,600 bd eingestellt. Die Timer-Ausgänge J8,9/3,4 sind mit den Eingängen J39(Receive Clock) und J36(Tranceive Clock) verbunden.

J28 und J29 übernehmen die Pegelwandlung für die V24-Schnittstelle. Die parallelen Ausgänge der STI J6 sind frei und für Erweiterungen vorgesehen; die freien Parallelausgänge J2A bis J7A werden zum GDP-Teil geführt und später erläutert.

### Der Graphik-Teil

-----

Kern des Graphik-Teiles ist der Video-Prozessor GDP 9366 für 512x256 Bildpunkte oder dem GDP 9365 mit 512x512 Bildpunkten. Die beiden Prozessoren sind über die Brücke J2 auswählbar.

Standard ist die Bestückung mit dem GDP 9366. TERM1 ist auch schon für den GDP 9367 vorgesehen.

Der GDP J10 ist an den Datenbus und an die Adressbits A0...A3 angeschlossen. Über diese Bits können die 16 internen GDP-Register adressiert werden. Der oben erwähnte Adressdecoder J2/7 wird, nach Verundung mit M1 über J12 (0-aktive Signale, daher UND) zum E\*-Eingang J10/17 geführt.

Der Arbeitstakt des Graphik-Teiles wird über den Quarz Q2 und dem Taktgenerator J23 erzeugt. Achten Sie bitte darauf, dass auf Grund der hohen Taktrate ein 7404 ohne LS eingesetzt wird!

Der 14 MHz Punkttakt führt direkt zum Schieberegister J25 (wird später behandelt) und zum Zähler J26, der die sonstigen im System benötigten Zeiten erzeugt. J26 zählt von 8 bis F und lädt sich nach Erreichen von "F" wieder selbst über J27/9,10,8.

Der Grundtakt steuert das Schieberegister J25 über J25/7. Weiter führt er zum Video-Mischer J12. Dadurch ergibt sich ein Punkttakt von 71 ns.

Der Grundtakt, der durch den Zähler J26 durch max. 8 geteilt wird, ergibt einen Systemtakt von 1.75 MHz, der am Ausgang J26/12 ansetzt und als CK zum GDP geführt wird.

Über J27 wird das CAS\*-Signal für die Speicher erzeugt.-

Der GDP J10 übergibt die 15 bit grosse Adresse eines Bildpunktes an seinen Ausgängen DAD0 (J10/37) ... DAD6 (J10/5) in Verbindung mit den Ausgängen MSLO (J10/6) bis MSL2 (J10/7). Die ersten Ausgänge führen direkt zu den Adresseingängen der Speicher; die Signale MSLO ... MSL2 zum Baustein J13, einem 3 zu 8-Decoder mit einigen Besonderheiten.

J13 decodiert die an seinen Eingängen (A,B,C) anstehende Information und schaltet jeweils einen Ausgang J13/11,9,8 usw. auf Low. Dadurch wird ein RAS\* erzeugt und die Adressen dem jeweiligen Speicherchip übergeben.

J27/1,2,3 verknüpft die Ausgänge ALL\* und MEMFREE\* des GDP. Ein Enable (Freigabe) des J13 wird erreicht, wenn der Ausgang DW\* (J10/14) aktiv ist (über E1\*, J13/16) und wenn nicht beide Ausgänge ALL\* und MEMFREE\* Null sind (über J27/1,2,3).

Der POL-Eingang des Multiplexers, J13/12, schaltet alle Ausgänge um. Dies wird bei unserer Schaltung angewandt, um die Speicherinhalte wieder auslesen zu können. Zu diesem Zeitpunkt wären alle Ausgänge von J13 auf "1"; durch POL werden Sie zu "0", d.h. alle Speicher erhalten ein identisches RAS\*, das die Ausleseeadresse definiert. Dies wünscht der GDP durch seinen Ausgang ALL\*.

Bei MFREE\* können die Speicher rückgelesen werden, z.B. für eine Hardcopy. Diese Funktion ist allerdings beim TERM1 nicht realisiert. Die PAGE-Logik stellt J14 dar. TERM1 kann vier Seiten bedienen, wobei in eine Seite geschrieben und gleichzeitig eine weitere Seite angezeigt werden kann.

Damit es hier nicht zu Kollisionen kommen kann, dient J14, ein Zweifach-Multiplexer.

Zunächst sei der Jumper J2 auf der Stellung "9366". Damit ist nur der linke Teil von J14 interessant.

J14 erzeugt mit seinem Ausgang J14/7 das höchstwertige Adressbit A15, das abhängig von der gewählten Page ist. Eine weitere Logik (J11 über J26) sorgt dafür, dass diese Adresse auch richtig im Timing der Speicher erzeugt wird.

Abhängig von der Information an den Eingängen A und B (J14/14,2) wird einer der Eingänge 0...3 (J14/3,4,5,6) durchgeschaltet.

Die Eingänge J14/0,1 bestimmen die Seite, aus der gelesen werden soll, J14/2,3 definieren die Schreibseite.

Um Störungen zu vermeiden, darf in ein Bild nur dann geschrieben werden, wenn der Stahl ausserhalb des darstellbaren Bereiches liegt. Der GDP J10 teilt dies durch seinen BLK (Blank-) Ausgang J10/25 mit, der an einen Eingang von J14 gelegt ist. Der andere Eingang wird von einer Taktflanke, die mit J27 und J11 zwischen RAS und CAS liegt, belegt. Dadurch wird die Adresse einer aktuellen Seite nur zum erlaubten Zeitpunkt umgeschaltet, da die dynamischen Speicher die Adressinformation ja sequentiell angelegt bekommen.

Welche Page angesprochen ist, wird durch das höchstwertige Adressbit A15, das über J14/7 geschaltet wird, definiert.

Die Ausgabedaten der RAMs führen an die Eingänge des 8-bit-Schieberegisters J25. Jeweils zu Beginn eines neuen Taktschrittes werden die acht Punktdaten hier übernommen, gesteuert durch J12. Während der BLANK-Zeit, gesteuert durch J10/25 BLK über J12, wird die feste "1" des Serial Inputs übernommen. Eine "1" entspricht einem dunklen Bildpunkt.

Die hinausgeschobenen Daten (J25/13) werden mit dem Punkttakt verknüpft über J12/12,13, mit dem Synchronisationssignal über SYNC(J10/34), J13 und den Widerständen R7,R9 und R10 vermischt.

Hinweis: Bei manchen Monitoren kann R10 (1K0hm) weggelassen werden - dies erhöht den Signalpegel.

T1 verstärkt das Signal und legt es Z-Richtig mit 75 Ohm an den Ausgang BAS.

Funktion des Bausteines 9366

Dieses Kapitel benötigen Sie nur dann, falls Sie den GDP direkt ansteuern wollen. Sie verwenden dazu den Befehl G a b

Sonst können Sie überblättern und direkt zum Kapitel "Quick Reference" oder "Anwendungsbeispiele" übergehen.

Die Steuerung des TERM1 übernimmt der "Graphik Display Prozessor EF 9366" von Thomson-CSF.

Tabelle 1 zeigt die Registeradressen und die Bedeutung:

TABLE 1 - REGISTER ADDRESS

ADDRESS REGISTER					REGISTER FUNCTIONS		Number of bits
Binary				Hexa	Read R/W = 1	Write R/W = 0	
A3	A2	A1	A0				
0	0	0	0	0	STATUS	CMD	8
0	0	0	1	1	CTRL 1 (Write control and interrupt control)		7
0	0	1	0	2	CTRL 2 (Vector and symbol type control)		4
0	0	1	1	3	CSIZE (Character size)		8
0	1	0	0	4	Reserved		—
0	1	0	1	5	DELTA X		8
0	1	1	0	6	Reserved		—
0	1	1	1	7	DELTA Y		8
1	0	0	0	8	X MSBs		4
1	0	0	1	9	X LSBs		8
1	0	1	0	A	Y MSBs		4
1	0	1	1	B	Y LSBs		8
1	1	0	0	C	XLP (Light pen)	Reserved	7
1	1	0	1	D	YLP (Light pen)	Reserved	8
1	1	1	0	E	Reserved		—
1	1	1	1	F	Reserved		—

Reserved: These addresses are reserved for future versions of the circuit. In read mode, output buffers DD071 force a high state on the data bus.

TABELLE 1 Registeradressen

Detaillierte Beschreibung der einzelnen Register:

0 Status / Kommando

Dieses Register ist das Schlüsselregister zum Baustein. Es kann vom Rechner gelesen werden - hier meldet der 9366 seinen Status - oder beschrieben werden - hiermit übergibt man ein Kommando zum Baustein.



## STATUS REGISTER (LESEN )

-----  
Bits:

7	6	5	4	3	2	1	0	
I	I	I	I	I	I	I	I	....HIGH Ende einer Lichtgriffelsequenz
I	I	I	I	I	I	I	I	.....HIGH Vertikal Blank
I	I	I	I	I	I	I	I	.....HIGH Bereit für neues Kommando
I	I	I	I	I				LOW BUSY
I	I	I	I	I	I	I	I	.....HIGH Pen ausserhalb Anzeigebereich
I	I	I	I	I	I	I	I	.....HIGH Lichtgriffel verursachte IRQ (Falls freigegeben)
I	I	I	I	I	I	I	I	.....HIGH Vertical Blank verursachte IRQ (Falls freigegeben)
I	I	I	I	I	I	I	I	.....HIGH Bereit für neues Kommando ver- ursachte IRQ (falls freigeg.)
I	I	I	I	I	I	I	I	.....HIGH Bits 5,6,7 verodert; HIGH falls überhaupt IRQ vorhanden

-----  
KOMMANDO - REGISTER (Schreiben)  
-----

Dieses Kommando-Register ist nicht ganz einfach zu verstehen - es hat fünf Funktionen:

Wert	Bedeutung
-----	
00H-0FH	Kommandos, wie Bildschirm , Register löschen usw.
10H-17H	Grundvektorbefehle  Ein Vektor wird definiert durch den Betrag in den Registern DELTAX und DELTAY sowie durch das hier übergebene Vorzeichen
18H-1F	Vektoren mit Richtungsvorgabe  Ein Richtungscode (8 Richtungen) wird übergeben, das kleinere der Register DELTAX oder DELTAY ignoriert und der Vektor mit der Länge des grösseren DELTA-Registers gezeichnet
20H-7FH	ASCII-Zeichen werden ausgegeben, Code-Zuordnung siehe Tabelle 2

80H-FFH Kurzvektoren

In einem Byte ist ein Vektor ollständig definiert

TABELLE 2 zeigt den Überblick über das Kommando-Register:

TABLE 2 - COMMAND REGISTER

		0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1																																																																																																					
		0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1																																																																																																					
		0	1	1	0	0	1	1	0	1	1	0	1	0	0	1	1																																																																																																					
		1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0																																																																																																					
b7	b6	b5	b4	b3	b2	b1	b0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F																																																																																																
0	0	0	0	0	0	0	0	0	SPACE	0	Q	P		p	<p>SMALL VECTOR DEFINITION:</p> <table border="1" style="margin: 5px;"> <tr> <td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>1</td><td><math> \Delta X </math></td><td><math> \Delta Y </math></td><td colspan="4">Direction</td> </tr> </table> <p>Dimension</p> <table border="1" style="margin: 5px;"> <tr> <td><math>\Delta X</math> or <math>\Delta Y</math></td><td>Vector length</td> </tr> <tr> <td>0</td><td>0</td><td>0 step</td> </tr> <tr> <td>0</td><td>1</td><td>1 step</td> </tr> <tr> <td>1</td><td>0</td><td>2 steps</td> </tr> <tr> <td>1</td><td>1</td><td>3 steps</td> </tr> </table> <p>Direction</p>							b7	b6	b5	b4	b3	b2	b1	b0	1	$ \Delta X $	$ \Delta Y $	Direction				$\Delta X$ or $\Delta Y$	Vector length	0	0	0 step	0	1	1 step	1	0	2 steps	1	1	3 steps																																																																				
b7	b6	b5	b4	b3	b2	b1	b0																																																																																																															
1	$ \Delta X $	$ \Delta Y $	Direction																																																																																																																			
$\Delta X$ or $\Delta Y$	Vector length																																																																																																																					
0	0	0 step																																																																																																																				
0	1	1 step																																																																																																																				
1	0	2 steps																																																																																																																				
1	1	3 steps																																																																																																																				
0	0	0	0	1					:	1	A	Q	a	q								<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>																																																																																																
0	0	1	0	1					-	2	B	R	b	r															<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>																																																																																									
0	0	1	1	0					#	3	C	S	c	s																						<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>																																																																																		
0	0	1	1	1					\$	4	D	T	d	t																													<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>																																																																											
0	1	0	1	0					%	5	E	U	e	u																																				<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>																																																																				
0	1	1	0	0					&	6	F	V	f	v																																											<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>																																																													
0	1	1	1	0					/	7	G	W	g	w																																																		<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>																																																						
1	0	0	0	0					f	8	H	X	h	x																																																									<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>																																															
1	0	0	1	0					j	9	I	Y	i	y	<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>																																																																																																							
1	0	1	0	0					:	J	Z	j	z	<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>																																																																																																								
1	0	1	1	0					.	K		k																																																																		<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>																																								
1	1	0	0						<	L	\	l	;																																																																								<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>																																	
1	1	0	1						=	M		m	!																																																																															<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>																										
1	1	1	0						>	N		n	~																																																																																						<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>																			
1	1	1	1						?	O	=	o	⊗																																																																																													<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>												
1	1	1	1						/	O	=	o	⊗																																																																																																				<p>Vector generation (for b2, b1, b0) and Small Vector Definition (for b2, b1, b0) are detailed in the table above.</p>					

TABELLE 2 Übersicht über das Kommando-Register

Bit Bedeutung

- 0 Im Register CTRL1 wird das BIT1 gesetzt; der "Schreibstift=Pen" wird angewählt  
Dieses Bit bzw. Kommando ist vor jedem Schreiben zu geben
- 1 Auswahl zwischen PEN und "Radiergummi":  
"1" PEN  
"0" ERASER (Gummi)



- 2 Im Register CTRL1 wird Bit 0 gesetzt - der PEN oder Eraser wird abgesenkt
- 3 Im Register CTRL1 wird das BIT 1 gelöscht - Pen oder Eraser werden angehoben
- 4 Bildschirm löschen
- 5 Register X und Y = 0
- 6 X,Y=0 und Bildschirm löschen
- 7 Alle Register (ausser XLP,YLP) zu 0 setzen, Bildschirm löschen
- 8 Lichtgriffel initialisieren (WHITE aktivieren, Bildschirm blinkt 1 Zyklus weiss)
- 9 Lichtgriffel initialisieren
- A 5x8-Block-Zeichnen. Die Grösse des Blocks ist von Register CSIZE abhängig
- B 4x4-Block-Zeichnen
- C Bildschirm scannen - Pen oder Eraser wie CTRL1
- D X=0
- E Y=0
- F Direkter Bildzugriff im nächsten freien Zyklus

Diese Befehle gelten, falls das erste Halbbyte des Register "0" ist, also für die Ausgaben "00H" bis "0FH".

#### Das Zeichnen von Vektoren

-----

Grundsätzlich ist ein Vektor durch seinen Startpunkt und seine Projektion auf die X- und Y-Achsen beschrieben.

Der Startpunkt wird durch den Inhalt der Register X und Y definiert

Die Projektionen auf die Achsen werden durch die Inhalte der Register DELAX und DELTAY übergeben. Das Vorzeichen wird erst im Kommando übergeben.

Während des Plottens werden die einzelnen Bildpunkte durch das X und Y-Register adressiert. Nach einem Plot zeigen diese Register auf den Endpunkt des Vektors.

Damit die Angelegenheit etwas komplizierter wird ( und um die Flexibilität in der Programmierung zu erhöhen ), gibt es 128 verschiedene Kommandos, Vektoren zu zeichnen. Wir wollen nun versuchen, Licht in das Dunkel zu bringen.

Die Vektorbefehle können in vier Gruppen aufgeteilt werden:  
 -Grundvektorbefehle  
 -Vektorbefehle parallel zur X oder Y-Achse  
 -Vektoren mit besonderer Richtungsvorgabe  
 -Kurzvektoren.

Grundvektorbefehle sind die allgemeinste Art der Vektordarstellung. Hier wird der Startpunkt des Vektors im Registerpaar X und Y angegeben; die Länge des Vektors (projiziert) in den Registern DELTAX und DELTAY. Das Vorzeichen dieser Register wird beim Befehl "Zeichne Grundvektor" übergeben.

Vektorbefehle parallel zur X oder Y-Achse sind ein Sonderfall der Grundvektorbefehle. Ein Register (DELTAX oder DELTAY) wird hierbei zu Null angenommen; im Befehl wird das Vorzeichen des anderen Registers übergeben.

Vektoren mit besonderer Richtungsvorgabe können nicht nur parallel zu den Achsen, sondern auch im 45-Grad - Winkel gezeichnet werden. Die Richtung wird hierbei im Befehl übergeben. Das kleinere der beiden Register DELTAX oder DELTAY wird ignoriert, da der Vektor ja durch die gegebene Richtung und eine (projizierte) Länge bereits vollständig beschrieben ist.

#### Grundvektorbefehle

```
-----
0 0 0 1 0 X X 1
      I I
      I I..... Vorzeichen von DELTAX  0="+"
      I I..... Vorzeichen von DELTAY  1="-"
```

#### Vektorbefehle parallel zu einer Achse

```
-----
0 0 0 1 0 X X 0
      I I
      I I..... 00 DELTAY=0,DELTAX>0
                  01 DELTAX=0,DELTAY>0
                  10 DELTAX=0,DELTAY<0
                  11 DELTAY=0,DELTAX<0
```

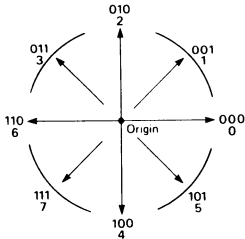
#### Vektoren mit besonderer Richtungsvorgabe



```

0 0 0 1 1 1 X X X
      I I I
      I I I...
    
```

Richtung, in folgendem Diagramm festgehalten



Das kleinere der Register DELTAX oder DELTAY wird ignoriert, der Vektor entsprechend der Richtung mit der Länge des grösseren Registers gezeichnet:

Kurzvektoren

```

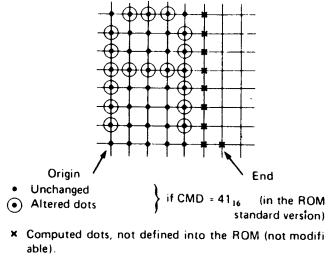
-----
1 x x x x x x x
  I I I I I I I
  I I I I I-I-I.....   Richtungscode
  I I I I
  I I I-I-I.....       DELTAY von 0..3 ohne Vorz.
  I I
  I-I-I.....           DELTAX von 0..3 ohne Vorz.
    
```

Da die gesamte Information über einen (Kurz-)Vektor in einem Byte übergeben werden kann, eignen sich die Kurzvektoren besonders gut, Umrisszeichnungen oder bewegte Bilder auszugeben. Die Vektoren können aufeinanderfolgend in einer Tabelle abgelegt werden und wie Texte ausgegeben werden.

Falls das Bit B7 (das höchstwertige) "0" ist und B6 bis B4 ungleich Null sind, so wird über das Kommandoregister ein ASCII-Zeichen übergeben. Dieses Zeichen wird an der Stelle X,Y mit der im CSIZE - Register angegebenen Grösse und der in CTRL2 definierten Richtung angezeigt.



Zeichen werden in einer 5 x 8-Matrix ausgegeben. Nach der Ausgabe eines Zeichens wird X um 6P erhöht. Dies verdeutlicht das folgende BILD:



Jeder der ausgegebenen Bildpunkte kann durch einen Block, der P x Q gross ist, ersetzt werden. P und Q können von 1 bis 15 reichen und werden in CSIZE übergeben.

Die Zeichen liegen von 20H bis 7FH und entsprechen den 96 Standard (USA) ASCII-Zeichen. Ein 97 stes Zeichen (0AH) erzeugt einen 5P x 8Q-Block und kann dazu verwendet werden, andere Zeichen zu löschen

Das 98 ste Zeichen erzeugt ein 4P x 4Q-Feld ohne Zwischenraum zum nächsten Zeichen. Mit diesem Zeichen können grobe graphische Zeichnungen (z.B. Balkendiagramme) erzeugt werden.

Ein Zeichen kann auf zweierlei Arten gelöscht werden: Entweder mit dem Zeichen 0AH, oder indem man das gleiche Zeichen (mit dem gleichen Startpunkt X, Y) und eingeschaltetem Eraser überschreibt.

Hinweis: Das Blank (20H) löscht nicht, sondern positioniert nur den X-Wert ein Zeichen weiter.

1 CTRL1 Steuerregister 1

-----  
TABELLE 3 zeigt den Aufbau des Registers CTRL1



TABELLE 3 Aufbau des Registers CTRL1

Einige Bits des Registers sind redundant, d.h. sie können auch über das Kontrollregister gesetzt bzw. gelöscht werden. Es sind dies die beiden (wichtigsten) niederwertigen Bits B0 und B1.

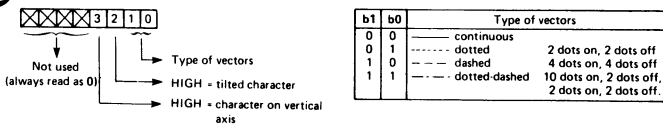
7	6	5	4	3	2	1	0		
I	I	I	I	I	I	I	I	....	HIGH=Pen unten LOW=Pen oben
								.....	HIGH=Pen LOW=Gummi
I	I	I	I	I	I	I	I	.....	HIGH=Schnell schreiben ohne Ausgangssignal
I	I	I	I	I	I	I	I	I	HIGH Geschlossene Bildfläche, d.h. es wird auch geschrieben, wenn MSB X,Y ausserhalb der Bildfläche sind
I	I	I	I	I	I	I	I	I	IRQ-Freigabe für Lichtgriffel
I	I	I	I	I	I	I	I	I	IRQ-Freigabe bei VB
I	I	I	I	I	I	I	I	I	IRQ-Freigabe bei "Bereit"
I	I	I	I	I	I	I	I	I	Nicht verwendet (0)

2 CTRL2 Steuerregister 2

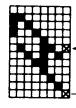
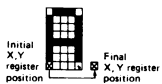
Steuerrigister 2 steuert die Art der gezeichneten Vektoren (Durchgezogen oder gepunktet oder strich-punktiert) sowie die Art der Zeichendarstellung.

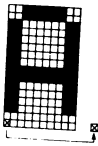
TABELLE 4 zeigt die Möglichkeiten des Registers 2

CONTROL REGISTER 2 (Read/Write)

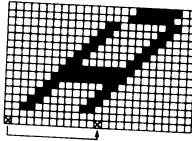


Types of character orientations





b3 = 0, b2 = 0  
CSIZE = 22<sub>16</sub>



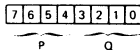
b3 = 0, b2 = 1  
CSIZE = 22<sub>16</sub>

3 Register CSIZE ZEICHENGRÖSSE

In diesem Register wird die Größe der darzustellenden Zeichen übergeben. Die Größe ist in X und Y-Richtung in 16 Schritten wählbar.

TABELLE 6 zeigt die Übergabe.

C-SIZE REGISTER (Read/Write)



P : Scaling factor on X axis  
Q : Scaling factor on Y axis

P and Q may take any value between 1 and 16. This value is given by the leftmost or rightmost 4 bits for P and Q respectively. Binary value (0) means 16.

TABELLE 6 CSIZE-Register

5,7 Register DELTAX und DELTAY

Diese Register werden bei den Grundvektorbefehlen verwendet und bedeuten die Projektion der Vektorlänge auf die jeweilige Achse. Ihre Bedeutung erhalten diese Register erst bei der Ausgabe von Grundvektoren. Im Befehl wird dann auch das Vorzeichen übergeben.

Mit DELTAX=DELTAY=0 können einzelne diskrete Punkte, deren Lage durch die Register X und Y definiert sind, ausgegeben werden.

8,9,A,B Register X und Y



Diese Register beinhalten den Startpunkt für jede Operation (Vektor Zeichnen oder Schriftausgabe)

Die Übergabe erfolgt in 12 Bit; damit ergibt sich ein virtueller Raum von 4096 x 4096 Bit, der vom Baustein auch verwaltet wird. 512 (x) x 256 (y) Bit werden angezeigt.

C,D Lichtgriffelregister  
-----

Hier wird vom Baustein der aktuelle Stand des Lichtgriffels übergeben. Diese Funktion ist im TERM1 nicht vorgesehen.

Anwendungsbeispiele

Wir haben zehn kleine Programme zusammengestellt, damit Sie sich mit dem Gerät und der Art der Programmierung etwas vertraut machen können.

Arbeiten Sie diese Beispiele durch, tippen Sie sie in den Rechner ein und verwenden Sie die Befehlsübersicht dazu!

Dies Beispiele sind im Microsoft MBASIC geschrieben - Das Graphiksystem ist in diesen Beispielen an den Druckerausgang angeschlossen und deshalb mit LPRINT-Befehlen anzusprechen.

Sollte Ihr Graphiksystem als Monitor geschaltet sein, so ersetzen Sie einfach alle "LPRINT"-Befehle durch "PRINT"-Befehle.

**Beispiellist** schon etwas trivial:Es schreibt das Wort "TEST" auf den Bildschirm.

```

5)TYPE DEMO1.BAS
10 'DEMO1 GES V.1.00
20 'SCHREIBT LEDIGLICH "TEST" AUF DEN BILDSCHIRM
100 LPRINT "TEST"
110 END

```

**Beispiell**

Im Beispiel 2 wird schon mehrmals eine Zeile angesprochen - hier wird das Scrollen gezeigt.

```

3)TYPE DEMO2.BAS
10 'DEMO2 GES V.1.00
20 'SCHREIBT SCHON 30 MAL TEST 1 ... TEST30 AUF DEN BILDSCHIRM
30 'HIER WIRD DAS SCROLLEN GEZEIGT
40 '
100 FOR I=1 TO 30
110 LPRINT "TEST":I
120 NEXT I
130 END

```

**Beispiel 2 zeigt das Scrollen**

im Beispiel 3 ist die Auswahl des Zeichensatzes (Deutsch oder Amerikanisch) erläutert. Beim Arbeiten mit deutschen Umlauten wird die Verarbeitungsgeschwindigkeit erheblich geringer, da hier nicht mehr auf den eingebauten Zeichengenerator zurückgegriffen werden kann, sondern die Umlaute und das scharfe "ö" Punkt für Punkt geplottet werden.

```

3)TYPE DEMO3.BAS
10 'DEMO3 GES GMBH V 1.00
20 'LÖSCHEN EINER SEITE UND UMSCHALTEN ASCII - DEUTSCHE UMLAUTE
30 '
40 ESCAPE=CHR$(27) 'ESCAPE
100 LPRINT CHR$(2#1) 'BILDSCHIRM NUN GELÖSCHT
110 '
120 GOSUB 1000 'ZUNÄCHST ASCII - GENT SCHNELL
130 '

```

```

140 LPRINT ESC#;" : 1"
150 GOSUB 1000
151
160 END
1000 FOR I=1 TO 24
1010 LPRINT "VIELE SCHLAUE FÜCHSE KAUFEN JEDE WOCHE WEITERE VIER BEQUEME PELZE: Ä Ö Ü Æ"
1020 NEXT I
1030 RETURN

```

### Beispiel 3 zeigt den Zeichensatz

Im Beispiel 4 zeigen wir einige Anwendungen der Steuerzeichen im Terminal-Betrieb. Zunächst wird der Bildschirm vollgeschrieben, dann kann der Cursor über das Bild gesteuert werden. Hier können auch Zeilen gelöscht oder eingefügt werden. Welche Tasten was bewirken, geht aus dem Programm hervor:

```

B>TYPE DEM04.BAS
10 'DEM04 GES GMBH V. 1.00
20 'CURSORSTEUERUNG
30 '
40 ESC#=CHR$(27)
100 LPRINT CHR#(&H1A)
110 '
120 GOSUB 1000
130 '
140 LPRINT "CURSORSTEUERUNG: 8 NACH OBEN, 6 NACH RECHTS, 2 NACH UNTEN UND 4 NACH LINKS"
145 LPRINT "7 ZEILE EINFÜGEN, 3 ZEILE LÖSCHEN, 0 ENDE"
150 GOSUB 2000
160 END
1000 FOR I=1 TO 20
1010 LPRINT "VIELE SCHLAUE FÜCHSE KAUFEN JEDE WOCHE WEITERE VIER BEQUEME PELZE: Ä Ö Ü Æ"
1020 NEXT I
1030 RETURN
2000 'EINHOLEN
2010 A$=INPUT$(1)
2020 IF A$="8" THEN LPRINT CHR#(&HB);
2030 IF A$="6" THEN LPRINT CHR#(&HC);
2040 IF A$="2" THEN LPRINT CHR#(&H16);
2050 IF A$="4" THEN LPRINT CHR#(&HD);
2052 IF A$="7" THEN LPRINT ESC#;"E";
2054 IF A$="3" THEN LPRINT ESC#;"R";
2060 IF A$="0" THEN RETURN
2070 GOTO 2010

```

### Beispiel 4: Cursorsteuerung

Nun aber zur Graphik!. Die Graphik-Betriebsart erreicht man durch die Sequenz: Escape Escape G, wie in Zeile 50 gezeigt. Das einfache Beispiel zeichnet einige Linien.

```

B>TYPE DEMOS.BAS
10 'DEMOS GS GMBH V 1.00
20 'ERSTE GRAPHIK
30
40 ESC$=CHR$(27)
50 LPRINT ESC$;ESC$;"G"          'UMSCHALTEN IN GRAPHIK
60 LPRINT "Z"                    'LÖSCHEN BILDSCHIRM
70 LPRINT "M0,0"                 'ZUM PUNKT 0,0 FAHREN
80 LPRINT "D0,100"              'NACH 0,100 LINIE ZIEHEN
90 LPRINT "D100,100"            'USW
100 LPRINT "D50,150"
110 LPRINT "D0,100"
120 LPRINT "M100,100"
130 END
140 LPRINT "D0,0"

```

### Beispiel 5 : Einfache Graphik

Mit dem Befehl "G" kann man dem Graphik Display Prozessor direkte Steuerbefehle übermitteln. Dies ist im entsprechenden Kapitel erklärt. Wir wenden hier im Beispiel 6 und 7 folgende Steuermöglichkeiten an:

- Umschalten auf Lösch-Darstellung, um Linien zu löschen. (Beispiel 6)
- Verändern der Buchstabengröße und Richtung (Beispiel 7)

```

B>TYPE DEM06.BAS
5 'DEM06
10 'ZEICHEN UND LÖSCHEN VON LINIEN
20 'ERSTE GRAPHIK
30
40 ESC$=CHR$(27)
50 LPRINT ESC$;ESC$;"G"          'UMSCHALTEN IN GRAPHIK
60 LPRINT "Z"                    'LÖSCHEN BILDSCHIRM
70 LPRINT "M 0,0"
80 LPRINT "D100,100"
90 LPRINT "D0,100"
100 LPRINT "D0,0"                'DREIECK ZEICHNEN
105 FOR I=1 TO 1000:NEXT I        'VERZÖGERUNG
110 LPRINT "G 1,1"               'GDP DIREKT KOMMANDO-
111                               'ALLE FOLGENDEN DRAW-
112                               'KOMMANDOS LÖSCHEN NUN
120 LPRINT "D100,100"
130 LPRINT "G 1,3"               'WIEDER NORMAL
140 LPRINT "D511,0"
150 END

```



**Beispiel 6 Zeichnen und Löschen von Linien**

```

10 'DEM07 GES GMBH GRAF
20 'BUCHSTABENDARSTELLUNG
30 '
35 LPRINT CHR$(27);CHR$(27);"G":LPRINT "Z"
36 LPRINT "MO,0"
40 LPRINT "BA"
50 LPRINT "G3 #33"
60 LPRINT "BA"
70 LPRINT "G2 #04"          'KURSIV
80 LPRINT "BA"
90 LPRINT "M300,0"
95 LPRINT "G3 #FF"        'MAX GRÖSSE
100 LPRINT "G2 #08"       'VERTIKAL
110 LPRINT "BA"

```

**Beispiel 7 Buchstabengrößen und Richtung ändern**

Unser Graphik-System verfügt über vier unabhängige Bildschirmseiten. Unabhängig deshalb, da man in eine Bildschirmseite eintragen kann und gleichzeitig eine andere Bildschirmseite anzeigen kann. Beispiel 8 zeigt diese Möglichkeit.

```

B>TYPE DEM08.BAS
5 'DEM08 GES GMBH GRAF
10 'SEITENUMSCHALTEN
20 '
30 '
40 ESC#=CHR$(27)
50 LPRINT ESC#;ESC#;"G"          UMSCHALTEN IN GRAPHIK
60 LPRINT "Z"
65 LPRINT "MO 0;G2 0; X 0"      KOMMANDOS SHACHTELN MÖGLICH!
66 LPRINT "P 0"                SCHREIB=LESESEITE=0
70 LPRINT "R511 255"
75 LPRINT "M10 200"
80 LPRINT "G3 #22"
90 LPRINT "BDIES IST DIE SEITE 0: SIE IST IMMER NACH "
100 LPRINT "M 10 180"
110 LPRINT "BDEM EINSCHALTEN VOREINGESTELLT"
115 FOR I=1 TO 500:NEXT        'WARTEN
120 LPRINT "P 5"              'SCHREIBSEITE=1
121                            'LESESEITE =1 EINSTELLEN
130 LPRINT "M 10 160"
140 LPRINT "BUND DAS IST SEITE 1"
150 FOR I=1 TO 500:NEXT
160 LPRINT "P 10"              'SEITE 2:2*4+2=10
170 LPRINT "M10 140"
180 LPRINT "BSEITE 2"
185 FOR I=1 TO 500:NEXT

```

```

190 LPRINT "G3 #11"          'SCHRIFT KLEINER
195 LPRINT "P 15"          'SEITE 3: 3*4*3
200 LPRINT "M10 100"
205 LPRINT "WAEHREND SIE HIER LESEN, WIRD IN SEITE 0 GEZEICHNET"
210 LPRINT "P 3"          'SCHREIBSEITE=0, LESE=3
215 LPRINT "M 5 5"
220 FOR IX=10 TO 100 STEP 5
230 FOR IY=10 TO 100 STEP 10
235 LPRINT "R ":IX:" ":IY
240 NEXT IY
250 NEXT IX
260 LPRINT "P0"          'JETZT ZEIGEN!
265 FOR I=1 TO 500: NEXT
270 LPRINT "X 50"          'UMSCHALTEN
280 FOR I=1 TO 5000: NEXT I
290 LPRINT "X 1"          'SCHNELL UMSCHALTEN
300 FOR I=1 TO 3000: NEXT
310 LPRINT "X 20"
999 END

```

### Beispiel 8 Vier Bildschirmseiten

DAS Graphik-System verfügt über eingebaute Funktionen zum Zeichnen von:

- Kreisen, Ellipsen und Kreissegmenten
- Dreiecken
- Polygonen
- Rechtecken

Beispiel 9 gibt hierüber Auskunft.

```

B>TYPE DEMO9.BAS
10 'DEMO9 GES GMBH GRAF 15.8.83
20 'KREISE UND KREISSEGMENTE
30 'SOWIE DREIECKE
40
50 ESC%=CHR$(27)
60 LPRINT ESC%:ESC%:"6"          'UMSCHALTEN IN GRAPHIK
70 LPRINT "Z"
80 LPRINT "M250 125"
90 LPRINT "0 30 20 0 0"          'KREIS, X=30, Y=20,
100                               'STARTWINKEL=0 GRAD,
110                               'ENDWINKEL=0 GRAD
120 GOSUB 260
130 LPRINT "M 200 100"
140 LPRINT "0 30 20 0 0 1"          'KREIS DTD, BEFÜLLT
150 GOSUB 260
160 LPRINT "M 50 175"
170 LPRINT "0 100 90 110 140 1"          'KREISSEGMENT VON 110 GRAD
180                               'BIS 140 GRAD, BEFÜLLT
190 GOSUB 260
200 LPRINT "Z"
210 LPRINT "1 0 0 400 0 200 200"          'GEFÜLLTES DREIECK
220 GOSUB 260
230 LPRINT "Z"
240 LPRINT "L 0 0 100 0 130 100 50 100"          'MALT BELIEBIGES POLYGON
250 END
260 FOR I=1 TO 500:NEXT:RETURN

```

/ -35-



**Beispiel 9 Ellipsen, Kreise, Polygoe**

Im Beispiel 10 zeigen wir, was unseres Wissens kein System dieser Preisklasse kann: Das Definieren und Zeichnen von Figuren in Echtzeit.

Mit einem Befehl kann ein Fadenkreuz an jeder Bildschirmseite an jeder Position dargestellt werden. Mit einer einfachen ASCII-Zeichenkette kann die Figur des Fadenkreizes zu einer beliebigen Figur umdefiniert werden. Nun kann diese vom Anwender definierte Figur wievorher das Fadenkreuz am Bildschirm positioniert werden. Aber nicht nur das - die Figur kann auch durch einen Befehl vergrößert oder verkleinert werden!

Beispiel 10 zeigt diese Möglichkeit - das Fadenkreuz wurde zu einem Auto umdefiniert.

```

B>TYPE DEMO10.BAS
10 'DEMO 10 GES GMBH GRAF 15.8.83
20 'FADENKREUZ UND AHNLICHE FIGUREN
30 '
40 '
50 ESC#=CHR$(27)
60 LPRINT ESC#;ESC#;"G"           UMSCHALTEN IN GRAPHIK
70 LPRINT "Z"
75 LPRINT "WA"                    'FADENKREUZSYMBOL AUSWAHLEN
80 LPRINT "F 100 100 0"          'ZEICHNE FADENKREUZ
90 FOR IX=100 TO 300 STEP 20    'UND FAHRE RECHTS
100 LPRINT "F ";IX;" 100 0"
110 NEXT IX
120 '
150 FOR I=1 TO 500:NEXT
155 LPRINT "MO 92;R 511 2"
160 LPRINT "WA1XXDH1XXDH1XRHLXAAATTTTTTTTTVVVVVTTVPPPOYY1XXZTTTVN"
165 LPRINT "WC 2"                'VERGRÖSSERN UM 2
170 LPRINT "F100 100 0"
180 FOR IX=100 TO 500
190 LPRINT "F ";IX;" 100 0"
200 NEXT IX

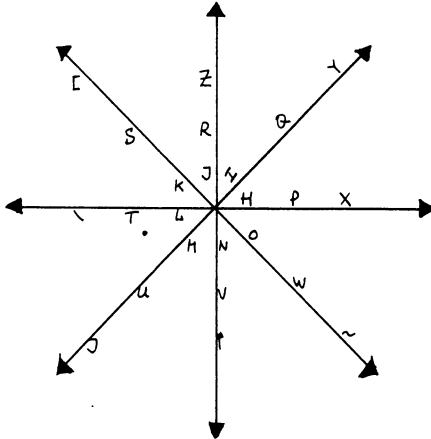
```

**Beispiel 10 - Fadenkreuz und Auto**

Wie entsteht der Befehl in Zeile 160?

Dazu dient das folgende Bild: Die Figur wird in sogenannte Kurzvektoren zerlegt; diese Kurzvektoren werden durch ein ASCII-Zeichen dargestellt. Eine "1" schaltet den "Zeichenstift" ein, eine "0" aus.

Beispiel: Schalte ein, gehe drei Schritte nach rechts "lX", dann einen Schritt nach oben "lXJ", dann einen Schritt nach links "lXJl", dann schalte aus "lXJLO" und gehe einen Schritt nach unten "lXJLON" und schalte wieder ein "lXJLONl" usw.



**BILD: Kurzvektoren und Ihre Verschlüsselung"**

Die im weiteren Anhang stehenden Beispielprogramme sollen nicht weiter erläutert werden. Hier ist für fortgeschrittene Programmierer die LOGO und Tectronix-Fähigkeit des Graphik-Systemes erläutert.

```

E>TYPE E:DEM01.BAS
10 CLEAR 2000
20 REM Beispiele fuer TERM 1
30 REM Rolf-Dieter Klein 830508
40 REM TERMINAL-TVI-950 Funktionen
50 PRINT CHR$(26); 'Clear Screen
60 FOR X=0 TO 79 'Alle Spalten
70 Y = SIN(X/79*2*PI)*11+11
80 Y = INT(Y)
90 PRINT CHR$(27);"=";CHR$(32+Y);CHR$(32+X);"*"
100 NEXT X
110 REM Wegschieben durch Scrollen
120 FOR I=1 TO 40:PRINT:NEXT I
130 REM
140 REM verschiedene Zeichensaetze
150 PRINT CHR$(27);"=";CHR$(32+5);CHR$(32)
160 PRINT CHR$(27);"z0" 'amerikan Zeichen
170 PRINT
180 FOR I=32 TO 127:PRINT CHR$(I);:NEXT I
190 PRINT
200 PRINT CHR$(27);"z1" 'deutsche Zeichen
210 FOR I=32 TO 127:PRINT CHR$(I);:NEXT I
220 PRINT
230 FOR I=1 TO 1000:NEXT I
240 REM Einfuegen, Loeschen etc.
250 FOR J=1 TO 2
260 PRINT CHR$(26);CHR$(27);"=$ "
270 PRINT "Testsatz fuer Einfuegen etc."
280 PRINT CHR$(11); ' cursor up
290 FOR I=1 TO 10:PRINT CHR$(27);"E";:NEXT I
300 FOR I=1 TO 10:PRINT CHR$(27);"R";:NEXT I
310 FOR I=1 TO 10:PRINT CHR$(27);"Q";:NEXT I
320 FOR I=1 TO 10:PRINT CHR$(27);"W";:NEXT I
330 PRINT 'cr ausgeben, wegen autocr von Basic
340 NEXT J

```

```

TYPE DEM02.BAS
10 REM Graphic Modus
20 REM Rolf-Dieter Klein 830508
30 PRINT CHR$(27);CHR$(27);"GZPO" 'Graphic Modus
40 PRINT "MO 0;R511 255" 'Rechteck-Rahmen zeichnen
50 PRINT "M50 200;G3 $22;BGraphic-Befehle"
60 PRINT "M200 50;O100 50 0 0 1"
70 PRINT "1300 0 511 20 400 200"
80 REM Symbol definieren
90 PRINT "WA";"HHHHIJKLLMMNO"
100 PRINT "Y1" 'zwei Seiten quasisimultan
110 PRINT "WC2 0" 'VERGROESSERUNG 2 DREHUNG 0
120 FOR X=0 TO 511
130 PRINT "F";X;INT(SIN(X/100*2*PI)*120+127);1
140 NEXT X
150 PRINT "A" 'in Alpha zurueck
999 PRINT "A"

```

```

TYPE DEM03.BAS
10 REM Statuszeile ueber Graphic Modus
20 REM Rolf-Dieter Klein 830508
30 PRINT CHR$(27);CHR$(27);"G" 'ohne Loeschen
40 PRINT "PO;MO 0;R511 15" 'mit Rahmen
50 PRINT "P5;MO 0;R511 15" 'beide Seiten
60 PRINT "PO;G3 $22;M40 -1;BStatus Zeile doppelt hoch"
70 PRINT "P5;M40 -1;BStatus Zeile hoch"
80 PRINT "A" 'wieder zurueck in Textmodus
90 LIST

```

```
TYPE DEMO4.BAS
10 REM Kleine LOGO-Demonstration
20 REM Rolf-Dieter Klein
30 PRINT CHR$(27);CHR$(27);"L" 'std mode
40 PRINT "M255 255 90" 'umgerechnet auf 512x256
50 FOR I=0 TO 360/10
60 GOSUB 130 'Muster 1
70 PRINT "P-10"
80 NEXT I
90 PRINT "TO" 'Turtle off
100 FOR I=1 TO 1000:NEXT I
110 PRINT "A"
120 END
130 REM Unterprogramm Figur
140 FOR N=1 TO 4
150 PRINT "F150;P";360/4
160 NEXT N
170 RETURN
```

```
E>TYPE DEMO5.BAS
10 REM Tektronix Demo
20 REM Rolf-Dieter Klein 830508
30 PRINT CHR$(27);CHR$(27);"TO 0 1 2 1"
40 FOR X=0 TO 1023 STEP 4
50 Y=(SIN(X/1023*2*PI)*500+512)
55 Y=INT(Y)
60 GOSUB 100 'Tektronix Ausgabe
70 NEXT X
80 FOR I=1 TO 1000:NEXT I
90 PRINT CHR$(27);CHR$(27);"A":END
100 REM x,y trennen und Ausgeben
105 PRINT CHR$(28);CHR$(7); 'Pointplot mode
110 PRINT CHR$(INT(Y/32)+32);CHR$(Y MOD 32 + 96);
CHR$(INT(X/32)+32);CHR$(X MOD 32 + 64)
999 RETURN
```

BEFEHLSBESCHREIBUNG

TERM1 Version 3.4 Rolf-D.Klein

## Tastatur-Eingabe STI-B

=====

0..7fh . Werden an die STI-A geleitet

80h Lokal-Modus ein -- gleichzeitig  
als No-Scroll-Befehl verwendbar.  
Term1 läßt sich verlustfrei  
mit dem nächsten Befehl wieder  
starten

81h Lokal-Modus aus -- Scroll weiter  
Befehl

Im Lokal-Modus können alle Befehle über die Tastatur  
gegeben werden, die sonst vom Hostrechner kommen.

## Befehle

=====

## ALPHA-MODUS

-----

Ist nach dem Einschalten aktiv

20h..7fh Codes der sichtbaren ASCII-Zeichen

07h	CTRL G	Bell wird an STI-B ausgegeben
08h	CTRL H	Backspace
09h	CTRL I	Cursor Right
0ah	CTRL J	Linefeed
0bh	CTRL K	Cursor Up
0ch	CTRL L	Cursor Right
0dh	CTRL M	Carriage Return
16h	CTRL V	Cursor Down
1ah	CTRL Z	Clear
1eh	CTRL ^	Home

## Escape Sequenzen

1bh 2eh x ESC . n Cursor Attribute setzen  
(n='0','1','2','3','4').  
1bh 2eh 3lh entspricht  
ESC . 1 und läßt Cursor  
blinken

/ -40-

lbh 3dh y x	ESC = r c	Cursor setzen (y,x =20h..7fh)
lbh 3fh	ESC ?	Cursor Adresse abfragen Ergebnis wird über STI-A in dem Format wie oben (r c) ausgegeben
lbh 44h x	ESC D c	c='L' Lokal-Modus sonst Voll-Duplex
lbh 51h	ESC Q	Zeichen bei Cursorposition einfügen
lbh 57h	ESC W	Zeichen bei Cursorposition löschen
lbh 45h	ESC E	Zeile bei Cursorposition einfügen
lbh 52h	ESC R	Zeile bei Cursorposition löschen
lbh 54h	ESC T	Zeile ab Cursorposition bis zum Ende löschen
lbh 74h	ESC t	Zeile ab Cursorposition bis zum Ende löschen
lbh 59h	ESC Y	Seite ab Cursorposition bis Ende löschen
lbh 79h	ESC y	Seite ab Cursorposition bis Ende löschen
lbh 7ah x	ESC z n	n='0' amerikanische Zeichensatz; n='1' deutscher Zeichensatz

## Dopplescape Sequenzen

CR = 0dh Carriage return

lbh lbh 54h 0dh ESC ESC T CR Tektronix-4010-Modus

lbh lbh 54h x x x x 0dh  
ESC ESC T dx dy sx sy f CR  
Tektronix-4010-Modus.  
dx dy werden auf die Koordinaten  
für das Bildfenster addiert  
(Verschiebung des Bildfensters,  
Bereich -1024..+1024, die Werte  
dx,dy,sx,sy und f können dezimal  
gegeben werden).  
sx und sy sind Zahlen, die  
den Abbildungsmaßstab bestimmen.  
Das Bild wird in der jeweiligen  
Richtung um 2 hoch sx bzw.  
2 hoch sy gestaucht.





Wenn  $sx=1$  und  $sy=2$  bei einer Auflösung von  $512 \times 256$  Bildpunkten gewählt sind, dann ist das gesamte Tektronix Bildfenster mit  $1024 \times 1024$  Bildpunkten sichtbar. Wird für  $f$  ein Wert angegeben (z.B. 1), dann ist ein Fadenkreuz an der aktuellen Cursorposition sichtbar. Wird  $f$  weggelassen ist kein Fadenkreuz vorhanden. Werden auch  $dx, dy, sx$  und  $sy$  weggelassen, wie im Kommando oben, so wird automatisch  $dx=0$   $dy=0$   $sx=1$   $sy=2$  gesetzt

1bh 1bh 4ch 0dh ESC ESC L CR	Logo-Modus. Um eine unverzerrte Darstellung bei einer Auflösung von $512 * 256$ Bildpunkten zu erreichen, wird die y-Koordinate jeweils durch 2 geteilt
1bh 1bh 4ch 30h 0dh ESC ESC L O CR	Logo-Modus. Ohne Division für eine Auflösung von $512 * 512$ Bildpunkten ist das Bild unverzerrt
1bh 1bh 47h ESC ESC G	Grafik-Modus
1bh 1bh 43h ESC ESC C	Duchgangs-Modus. An der STI-A ankommende Zeichen werden an die STI-B weitergereicht. Abschalten nur mit ESC ESC R, das aber auch noch übertragen wird. Dieser Befehl wird zum Beispiel für eine Farberweiterung oder einen Druckeranschluß benötigt
1bh 1bh 50h ESC ESC P	Parallel-Modus. Wie oben, jedoch mit Verarbeitung der Zeichen im Terminal
1bh 1bh 52h ESC ESC R	Rücksetzen

#### GRAFIK-MODUS

-----

Das Zeichen  $n$  steht für die Eingabe eines numerischen Wertes.  $n$  kann eine vorzeichenbehaftete Dezimalzahl sein oder eine sedezimale Zahl: 0, -50, \$FF, -\$4034. Mehrere numerischen Werte werden durch Leerzeichen getrennt.

CR ist das Abschlußzeichen. Es kann bis auf wenige Ausnahmen auch das Zeichen Strichpunkt (';') verwendet werden.

A	in den Modus zurückschalten, von dem aus die Grafik aufgerufen wurde
M n1 n2 CR	Positionieren auf $x=n1$ $y=n2$
D n1 n2 CR	Vektor nach $x=n1$ , $y=n2$ zeichnen

/ -42-



- m b1 b2 b3 b4 Wie M, jedoch mit binärer Übertragung und daher schnell.  
 $x = b1, b2$  (b1 ist höherwertiges Byte, b2 ist niederwertiges Byte)  
 $y = b3, b4$
- d b1 b2 b3 b4 Wie D, jedoch binär. Format wie bei m
- J n1 n2 CR Wie D, jedoch mit Angabe von relativen Koordinaten  
 $dx = n1$   $dy = n2$
- P n CR Seite asynchron anwählen.  
 $n = \text{Nummer der Schreibseite} * 4 + \text{Nummer der zu lesenden Seite}$   
 z.B. Schreibseite = 3, Leseseite = 1  
 $n = 3 * 4 + 1 = 13$ . Befehl: P 13 CR
- S n CR Seite synchron anwählen.  
 n wird wie oben bestimmt
- X n CR Die Seiten 0,1,2,3 werden zyklisch angezeigt, n gibt die Sichtdauer einer Seite an.  
 Die Anzeigedauer beträgt  $n * 20\text{ms}$   
 $n = 0$  beendet den Wechsel
- Y n CR Es werden jeweils nur zwei Seiten zyklisch angezeigt und zwar die Seiten 0 und 1, wenn eine dieser beiden Seiten als Leseseite definiert war, sonst 2 und 3
- C Löschen der aktuellen Schreibseite
- Z Löschen aller Seiten 0 bis 3.  
 Seite 3 ist danach als Lese- und Schreibseite angewählt
- G n1 n2 CR Befehl an GDP  
 $n1 = \text{Nummer des GDP-Ports (0..15)}$ ,  
 $n2 = \text{Datenwert an diesen Port}$
- B text CR Text (20h..7fh) an GDP-Port 0 senden.  
 Der Text wird ab aktueller Koordinate des GDP ausgegeben
- V binär 00h Binärdaten an den GDP-Port 0 senden.  
 Das ASCII-Zeichen NUL beendet die Übertragung
- O n1 n2 n3 n4 CR Ellipsenabschnitte zeichnen.  
 Mit  $n1$  wird die Länge der Halbachse in x-Richtung angegeben, mit  $n2$  die Länge der Halbachse in y-Richtung.  
 Mit  $n3$  wird der Startwinkel bezüglich der x-Achse in Grad angegeben. Mit  $n4$  der Endwinkel des Ellipsenabschnittes.  
 Der Ellipsenabschnitt wird von der aktuellen x,y-Koordinate bis zum Erreichen des Endwinkels gezeichnet. Der Ellipsenmittelpunkt wird aus der Startwinkel- und Halbachsenangabe vor Beginn des Zeichnens automatisch errechnet.



- O n1 n2 n3 n4 1 CR  
Wie oben oben, jedoch der vom Kurvenstück und den Radien zum Mittelpunkt begrenzte Raum gefüllt (Torte)
- R n1 n2 CR  
Rechteck ab aktueller x,y-Koordinate. n1=dx und n2=dy geben die Breite und die Höhe des Rechtecks an
- L n1 n2 n3 n4 ... nn nm CR  
Polygon zeichnen, mit absoluten Koordinaten. x0=n1 y0=n2 gibt die Startposition an, alle weiteren Paare geben die Eckpunkte des Polygons an. Der letzte Eckpunkt wird wieder mit dem Startpunkt verbunden
- I n1 n2 n3 n4 n5 n6 CR  
Dreieck gefüllt zeichnen  
x0 = n1 y0 = n2  
x1 = n3 y1 = n4  
x2 = n5 y2 = n6
- F n1 n2 n3 CR  
Fadenkreuz zeichnen, an Position x=n1 y=n2, auf Seite n3 (0..3). Altes Fadenkreuz wird gelöscht. Die Schreib- und Leseseite bleiben erhalten
- WA string CR  
Symbol für den Fadenkreuz-Befehl undefinieren.  
string ist eine Zeichenkette mit Zeichen im ASCII-Bereich (30h,31h,40h..5fh).  
0 (30h) = Schreibstift hoch,  
1 (31h) = Schreibstift runter.  
Der Code für die Schreibstiftbewegung berechnet sich wie folgt:  
Richtung + (8 \* Länge) + 40h  
Es stehen die Richtungen von 0 bis 7 (\* 45 Grad) zur Verfügung
- WA CR  
Symbol für den Fadenkreuzbefehl auf das Symbol "Fadenkreuz" zurücksetzen
- WB  
Fadenkreuzsymbol an aktueller x,y-Koordinate auf der aktuellen Schreibseite setzen
- WC n1 n2 CR  
Fadenkreuzsymbol vergrößern und drehen.  
n1 = Vergrößerungsfaktor (1..255)  
n2 = Drehung (0..7)
- WC CR  
Rückstellen auf n1 = 1,  
n2 = 0
- WD n1 n2 n3 ... nn CR  
Download. Es können die Daten n2 bis nn ab Adresse n1 in den Arbeitsspeicher der Term1 geladen werden
- WE n1 CR  
Programm auf Adresse = n1 starten

/ -44-



WF n1 CR           Byte mit Adresse n1 aus dem  
Speicher der Terml lesen.  
Ergebnis wird binär über den  
Port der STI-A übertragen

### Doppelescape

1bh 1bh 41h	ESC ESC A	Alpha-Modus (oder Logo,T4010)
1bh 1bh 43h	ESC ESC C	Durchgangsmodus
1bh 1bh 50h	ESC ESC P	Parallel-Modus
1bh 1bh 52h	ESC ESC R	Rücksetzen

### LOGO-MODUS

-----

A	Alpha-Modus
Z	Lösche Bildschirm, alte Koordinaten bleiben
M n1 n2 n3 CR	Turtle Positionieren für Initialisierung. x=n1 y=n2 Startwinkel=n3 , Angabe in Grad (0..359)
F n1 CR	Vorwärts (bei negativer Zahl Rückwärts). n1 ist die Schrittzahl
P n1 CR	Drehung der Schildkröte von dem aktuellen Winkel aus um den Winkel n1 (in Grad, + und - möglich).
D	Pen Down
U	Pen Up
T0	Turtle unsichtbar
T1	Turtle sichtbar
1bh 1bh 41h	ESC ESC A           Alpha-Modus
1bh 1bh 47h	ESC ESC G           Grafik-Modus aufrufen
1bh 1bh 43h	ESC ESC C           Durchgangsmodus
1bh 1bh 50h	ESC ESC P           Parallel-Modus
1bh 1bh 52h	ESC ESC R           Rücksetzen



## TEKTRONIX-4010-MODUS

Es sind implementiert:

Alpha-, Grafik-, Incremental-, Pointplot-Modus

Nicht implementiert:

Gin-Modus nicht implementiert

(Definitionen siehe Tektronix 4010-Manual)

## T4010-Alpha-Modus:

20h bis 7fh	Textzeichen im Tektronix-Alpha-Modus
08h	Backspace
0bh	Cursor Up
0ah	Linefeed
0dh	Carriage Return
1ch	Pointplot-Modus
ldh	Grafik-Modus
leh	Incremental-Modus
1bh 05h	Übertragen der Cursorposition im 1024 * 1024 Raster
1bh 0ch	Clear All
1bh 1ah	Gin-Modus -- Nicht implementiert bleibt in Alpha-Modus
1bh 1ch	Spezial Pointplot-Modus -- hier wie Pointplot-Modus

Doppel-ESC-Sequenzen nur vom T4010-Alpha-Modus aus wirksam

1bh 1bh 41h	ESC ESC A	Alpha-Modus
1bh 1bh 47h	ESC ESC G	Grafik-Modus aufrufen
1bh 1bh 43h	ESC ESC C	Durchgangsmodus
1bh 1bh 50h	ESC ESC P	Parallel-Modus
1bh 1bh 52h	ESC ESC R	Rücksetzen

## Incremental-Modus:

Zeichen

A E D F B J H I

Richtungsvektor.  
Jeweils in 45 Grad Schritten  
gegen den Uhrzeiger  
(A= 0 Grad ... I = 315 Grad)

P	Pen Down
20h	Pen Up
0dh	in Alpha-Modus zurück
1fh	in Alpha-Modus zurück
1bh 0ch	Clear und in Alpha-Modus zurück

/ -46-



## Pointplot-Modus:

wie beim Grafik-Modus, nur daß keine Vektoren sondern nur die Endpunkte gezeichnet werden

## Grafik-Modus:

20h bis 7fh Vektorkoordinaten

Die Koordinaten x und y werden in niederwertige (Low) und höherwertige (High) Teile zerlegt.

Lowx = 40h..5fh die unteren 5 Bit der Koordinate werden verwendet  
 Lowy = 60h..7fh die oberen 5 Bit der Koordinate werden verwendet  
 Highx,Highy = 20h..3fh

HIY LOY HIX LOX Übertragungsformat: Es werden maximal vier Koordinatenteile übertragen.  
 HIY LOX Sofern sich die Koordinatenteile nur teilweise ändern, können auch bleibende Koordinatenstücke nach dem links stehenden Schema fortgelassen werden  
 LOY LOX  
 LOY HIX LOX  
 LOX

zusätzlich bei T4014 (Auflösung 4096 x 4086)

HIY EXTRA LOY HIX LOX Extra wird ignoriert

07h Erster Vektor wird doch geschrieben

0dh Alpha-Modus

1ch Pointplot-Modus

1dh Vektor ungeschrieben

1eh Incremental-Modus

1fh Alpha-Modus

1bh 05h Grafik-Cursor-Koordinaten Ausgabe

1bh 0ch Clear und dann Alpha-Modus

1bh 1ah Gin-Modus -- hier Alpha-Modus

1bh 1ch Spezial-Pointplot -- hier Pointplot

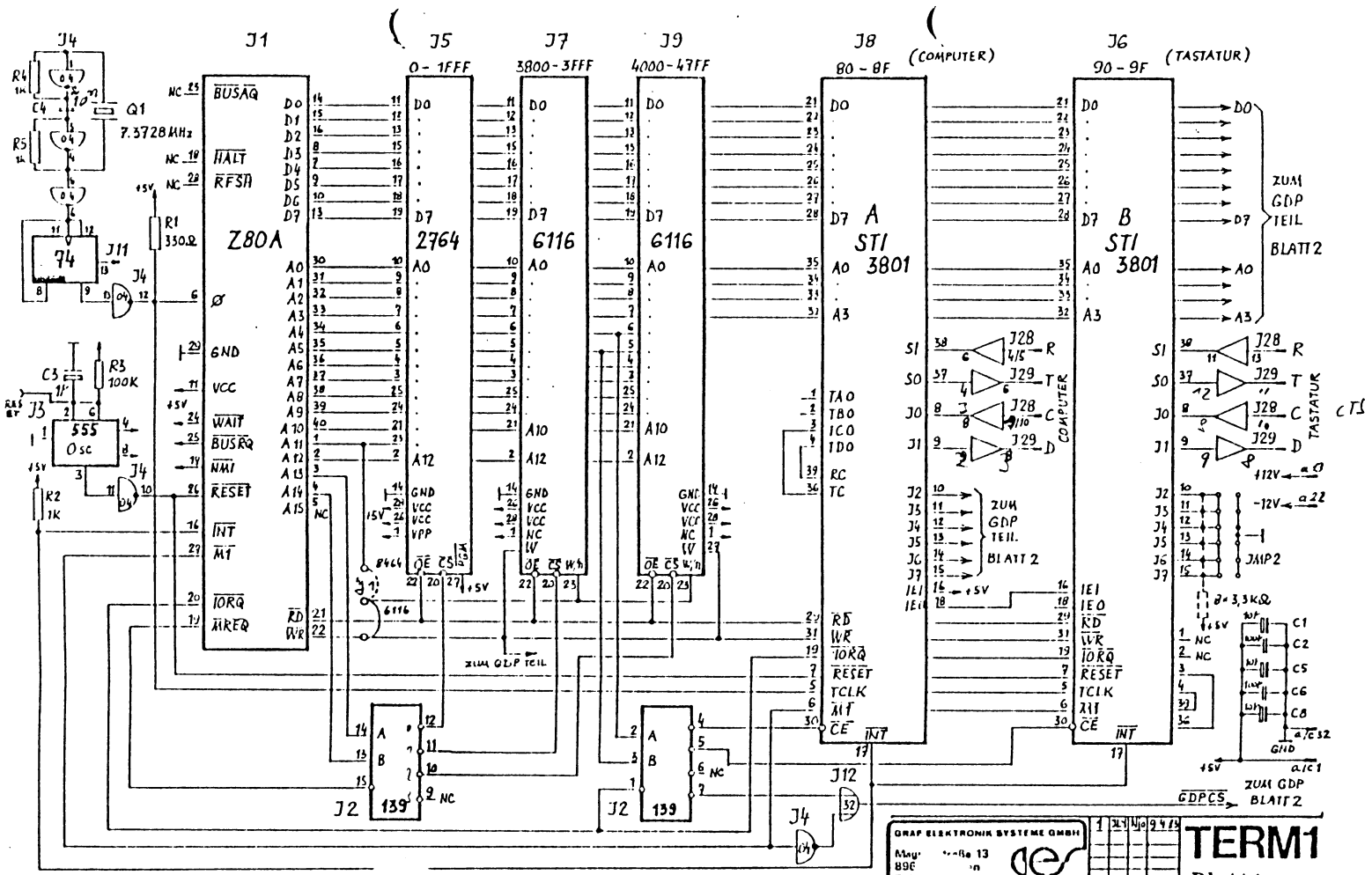


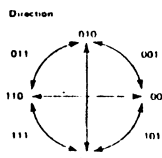
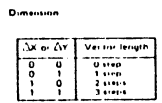
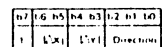
TABLE 1 - REGISTER ADDRESS

ADDRESS REGISTER					REGISTER FUNCTIONS		Number of bits
Binary					Read R/W = 1	Write R/W = 0	
A3	A2	A1	A0	Hexa			
0	0	0	0	0	STATUS	CAID	8
0	0	0	1	1	CTRL 1 (Write control and interrupt control)		7
0	0	1	0	2	CTRL 2 (Vector and symbol type control)		4
0	0	1	1	3	CSIZE (Character size)		4
0	1	0	0	4	Reserved		—
0	1	0	1	5	DELTA X		8
0	1	1	0	6	Reserved		—
0	1	1	1	7	DELTA Y		8
1	0	0	0	8	X MSBs		4
1	0	0	1	9	X LSBs		4
1	0	1	0	A	Y MSBs		4
1	0	1	1	B	Y LSBs		4
1	1	0	0	C	XLP (Light pen)	Reserved	7
1	1	0	1	D	YLP (Light pen)	Reserved	8
1	1	1	0	E	Reserved		—
1	1	1	1	F	Reserved		—

Reserved: These addresses are reserved for future versions of the circuit. In read mode, output buffers DO D7 force a high state on the data bus.

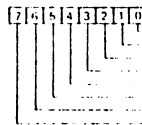
TABLE 2 - COMMAND REGISTER

b7	b6	b5	b4	b3	b2	b1	b0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0		
0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0		
0	0	0	0	0	0	0	0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F		
0	0	0	0	0	0	0	0	Vector generation (b7-b2) for small vector definition	SPACE	0	+	P	-	D	SMALL VECTOR DEFINITION	b7	b6	b5	b4	b3	b2	b1	b0		
0	0	0	0	1	Clear bit 1 of CTRL 1	Pen selection	1		A	Q	a	q	1	0		1	0	1	0	1	0	1	0	1	0
0	0	0	1	Eraser selection	Set bit 0 of CTRL 1	Pen/Eraser down selection	2		B	R	d	r	0	1		0	1	0	1	0	1	0	1	0	1
0	0	1	0	Clear bit 0 of CTRL 1	Pen/Eraser up selection	3	C		S	c	s	1	0	1		0	1	0	1	0	1	0	1	0	1
0	1	0	0	Clear screen	X and Y registers reset to 0	4	O		T	G	t	0	0	0		0	0	0	0	0	0	0	0	0	0
0	1	0	1	X and Y registers reset to 0	Clear screen, set CSIZE to code minimize	5	E		L	F	f	0	1	0		1	0	1	0	1	0	1	0	1	
0	1	1	0	X and Y reset to 0 and clear screen	Clear screen, set CSIZE to code minimize	6	F		V	v	v	1	0	1		0	1	0	1	0	1	0	1	0	1
0	1	1	1	All other registers reset to 0 (except XLP, YLP)	Light pen initialization	7	G		W	w	w	0	0	0		0	0	0	0	0	0	0	0	0	0
1	0	0	0	Light pen initialization (WRITE ignored)	Light pen initialization	8	H		X	x	x	1	1	1		1	1	1	1	1	1	1	1	1	1
1	0	0	1	Light pen initialization	Light pen initialization	9	I		Y	y	y	1	1	1		1	1	1	1	1	1	1	1	1	1
1	0	1	0	5 × 8 block drawing (size according to CSIZE)	5 × 8 block drawing (size according to CSIZE)	10	J	Z	z	z	0	1	1	1	1	1	1	1	1	1	1	1	1		
1	0	1	1	4 × 4 block drawing (size according to CSIZE)	4 × 4 block drawing (size according to CSIZE)	11	K	[	[	[	0	1	1	1	1	1	1	1	1	1	1	1	1		
1	1	0	0	Screen drawing	Screen drawing	12	L	]	]	]	0	1	1	1	1	1	1	1	1	1	1	1	1		
1	1	0	1	Pen or Eraser as defined by CTRL 1	Pen or Eraser as defined by CTRL 1	13	M	^	^	^	0	1	1	1	1	1	1	1	1	1	1	1	1		
1	1	1	0	X register reset to 0	X register reset to 0	14	N	~	~	~	0	1	1	1	1	1	1	1	1	1	1	1	1		
1	1	1	1	Y register reset to 0	Y register reset to 0	15	O	~	~	~	0	1	1	1	1	1	1	1	1	1	1	1	1		
1	1	1	1	Event image memory access request for the next frame cycle	Event image memory access request for the next frame cycle	16	P	~	~	~	0	1	1	1	1	1	1	1	1	1	1	1	1		



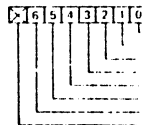
OTHER REGISTERS

STATUS REGISTER (Read only)



- bit 7: HIGH - light pen sequence ended
- bit 6: HIGH - vertical blanking (stem on pin VB) and not masked
- bit 5: HIGH - ready for a new command (LOW - busy)
- bit 4: HIGH - pen out of display window (logical OR of the 6 MSBs of the X and Y registers)
- bit 3: HIGH - light pen sequence ended (IRQ if enabled)
- bit 2: HIGH - vertical blanking IRQ (if enabled)
- bit 1: HIGH - ready for a new command (IRQ if enabled)
- bit 0: HIGH - logical OR of bits 4,5,6 - HIGH when IRQ output is low

CONTROL REGISTER 1 (Read/Write)



- bit 7: HIGH - pen down, LOW - pen up (control DW output)
- bit 6: HIGH - pen - LOW - eraser (control DIN output)
- bit 5: HIGH - high speed write (in video (BLK output is high, min. of memory refresh cycles)
- bit 4: HIGH - cyclic screen (memory display write even if bit 3 of the status register is high)
- bit 3: HIGH - enable end of the light pen sequence IRQ
- bit 2: HIGH - enable VB IRQ
- bit 1: HIGH - enable ready for a new command IRQ
- bit 0: Not used (0 for reading)

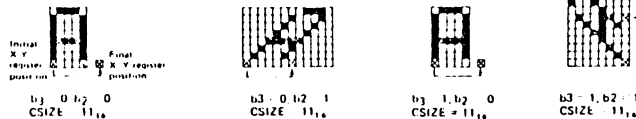
CONTROL REGISTER 2 (Read/Write)



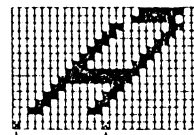
- bits 7-3: Not used (always read as 0)
- bit 2: Type of vectors
- bit 1: HIGH - tilted character
- bit 0: HIGH - character on vertical axis

b1	b0	Type of vectors
0	0	continuous
0	1	dotted 2 dots on 2 dots off
1	0	dashed 4 dots on, 4 dots off
1	1	dotted-dashed 10 dots on, 2 dots off
		2 dots on, 2 dots off

Types of character orientations

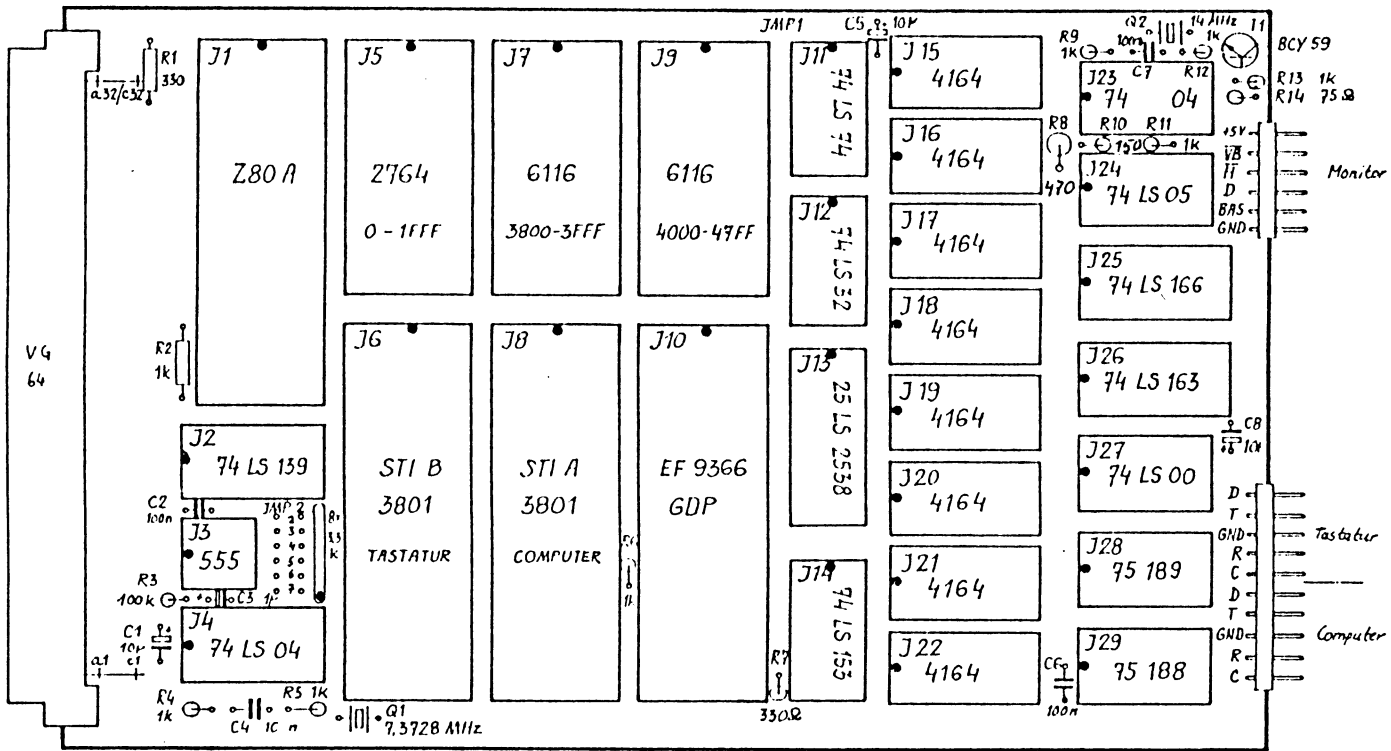


b3 = 0, b2 = 0  
CSIZE = 22<sub>10</sub>



b3 = 0, b2 = 1  
CSIZE = 22<sub>10</sub>



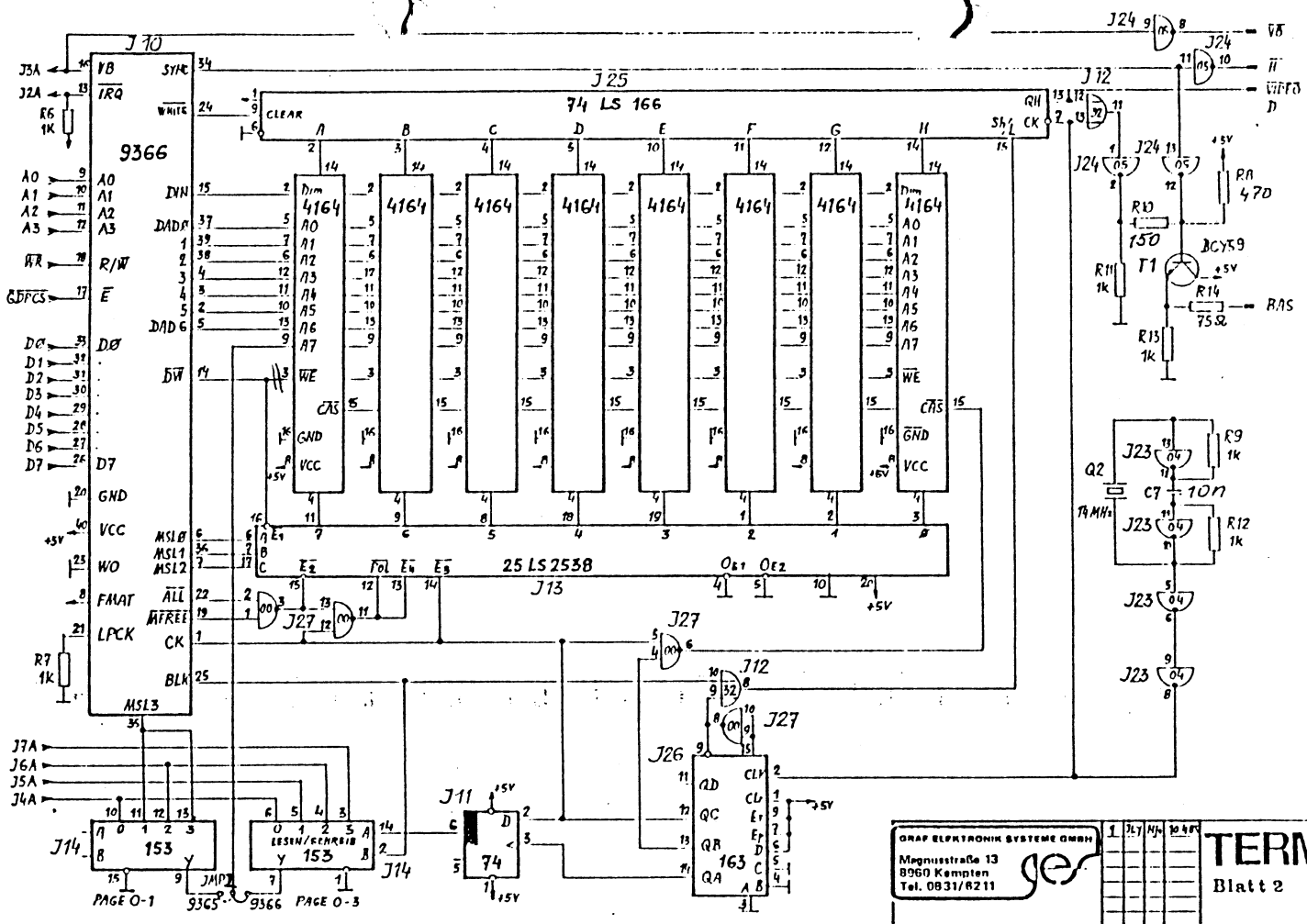


GRAF ELEKTRONIK SYSTEME GMBH  
 Magnusstraße 13  
 8960 Kempten  
 Tel. 08 31/62 11



1	157	164	164
2	157	164	164

TERM1  
 BP



## Zusatzblatt für TERM1 mit EF 9367 Prozessor

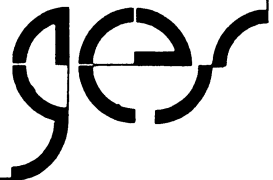
Die Verwendung des Grafik-Prozessors EF 9367 in der Term1 erfordert einige Änderungen gegenüber dem Einsatz des 9366. Die Durchführung geschieht folgendermaßen:

1. EF 9367 einsetzen
2. 12 MHz Quarz statt 14 MHz einlöten
3. Leiterbahn des EF 9367 Pin 19 auf der Lötseite auftrennen (x)
4. Brücke auf J27 Pin 1 nach Pin 2
5. Leiterbahn des EF 9367 Pin 8 auf der Bestückungsseite auftrennen. (f)
6. Brücke von EF 9367 Pin 8 nach GND. einlöten

Nähere Beschreibungen finden Sie im mc-Sonderheft und in der mc-Mikrocomputer- Zeitschrift 9/83

**Graf Elektronik Systeme GmbH**

Magnusstrasse 13  
8960 Kempten  
Telefon (0831)6211  
Teletex 831 804 = GRAF



## Zusatzblatt für Term 66/67

1. Die Kondensatoren C7 und C4 werden durch jeweils 10 nF ersetzt.
2. Der Netzwerkwiderstand  $8 \times 3,3 \text{ k}$  ist mit einem weißen Punkt markiert. Der Netzwerkwiderstand muß so bestückt werden, daß der weiße Punkt in Richtung des ICs 74 LS 04 zeigt.
3. Der Widerstand R8 wird durch einen 470 Ohm Widerstand und R 10 durch einen 150 Ohm Widerstand ersetzt.

**Graf Elektronik Systeme GmbH**

Magnusstrasse 13  
8960 Kempten  
Telefon (0831)6211  
Teletex 831 804 = GRAF

