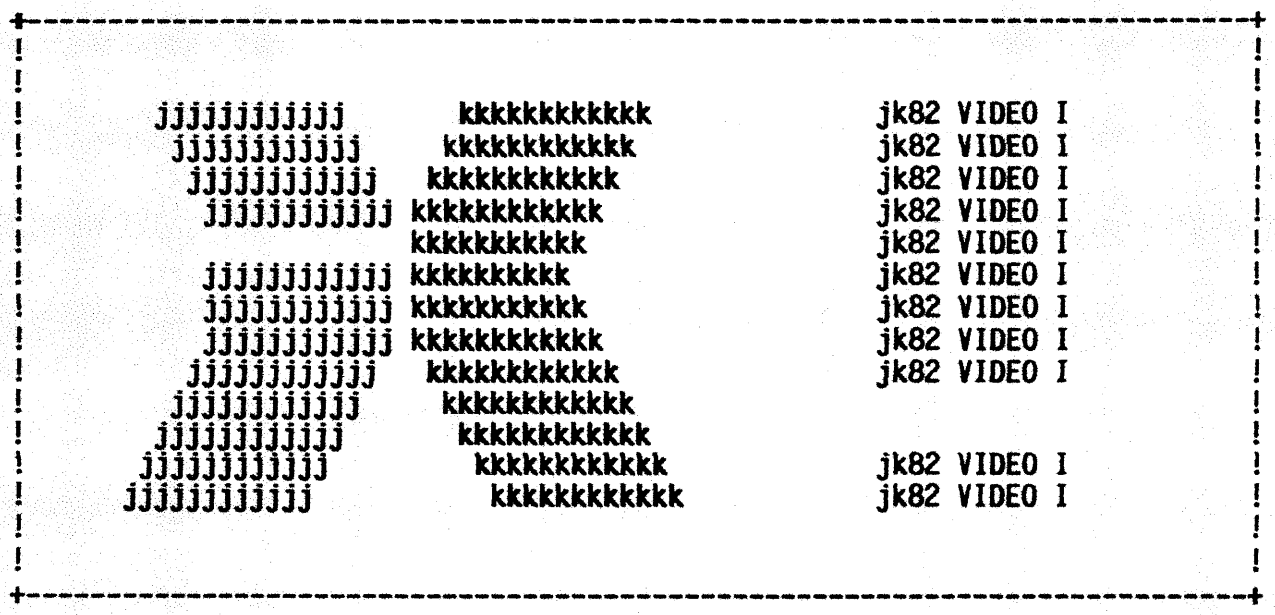
Janich & Klass
Computersysteme

jk 82 Video I

Anwender-Handbuch



Anwender-Handbuch zur jk82 VIDEO I

2. Auflage 1985

Dieses Handbuch ist gültig für die Version 1.1xx der Firmware zur jk82 VIDEO I.

Inhaltsverzeichnis Seite

1. Übersicht

Die Systemplatine jk82 VIDEO I ist ein intelligentes, universelles Video-Interface für alfanumerische und grafische Darstellung. Sie ist an jeden Z80-Rechner mit jk82-Bus anschließbar und wird über eine Parallelschnittstelle als I/O-Port angesprochen. Der parallele Datentransfer erlaubt eine hohe Übertragungsgeschwindigkeit und damit eine entsprechend schnelle Darstellung auf dem Monitor. Ein eigenes Z80 Subprozessorsystem garantiert hohe Flexibilität.

Als periphere Geräte lassen sich eine Tastatur (parallele oder serielle Datenübergabe) sowie ein Monitor mit BAS-Eingang anschließen. Die Darstellung auf dem Bildschirm ist flimmerfrei, da der Zugriff auf den Umlaufspeicher nur in den Austastlücken des BAS-Signals (horizontal und vertikal) erfolgt. Die Eingabe von der Tastatur wird durch das Video-Interface kontrolliert, bevor sie an den Rechner weitergegeben wird.

Im Alfa-Modus sind die folgenden 15 Bildschirmformate vom Hauptrechner aus anwählbar:

- 24 Zeilen a 80 Zeichen (alle Versionen)
  24 Zeilen a 96 Zeichen (alle Versionen)
  24 Zeilen a 132 Zeichen (alle Versionen)

- 25 Zeilen a 80 Zeichen (alle Versionen)
  25 Zeilen a 96 Zeichen (alle Versionen)
  25 Zeilen a 132 Zeichen (alle Versionen)

- 24 Zeilen a 40 Zeichen (Version 3 und 5)
  24 Zeilen a 48 Zeichen (Version 3 und 5)
  24 Zeilen a 66 Zeichen (Version 3 und 5)

- 12 Zeilen a 80 Zeichen (Version 4 und 5)
  12 Zeilen a 96 Zeichen (Version 4 und 5)
  12 Zeilen a 132 Zeichen (Version 4 und 5)

- 12 Zeilen a 40 Zeichen (Version 5)
  12 Zeilen a 48 Zeichen (Version 5)
  12 Zeilen a 66 Zeichen (Version 5)

Für jedes einzelne Zeichen können bei den ersten 6 der obigen Bildschirmformate folgende 4 Attribute programmiert werden: inverse Zeichendarstellung, halbe Helligkeit ('half intensity'), unterstrichene Darstellung ('underline'), blinkende Darstellung. Bei den restlichen 9 Bildschirmformaten steht das Attribut 'underline' nicht zur Verfügung.

Alle Attribute können beliebig miteinander kombiniert und für jedes Zeichen einzeln vereinbart werden, ohne daß ein Platz im Bildwiederholspeicher belegt wird. Zusätzlich besteht die Möglichkeit den ganzen Bildschirm invers darzustellen. Bei Änderung des Zeichengenerators kann statt 'underline' auch ein 2. Zeichensatz (z.B. griechische Buchstaben) selektiert werden.

VIDEO I speichert mehrere Bildschirmseiten, zwischen denen beliebig geblättert werden kann. Die Anzahl der gespeicherten Bildschirmseiten hängt vom gewählten Bildschirmformat ab. Im 24x80-Modus werden 8, im 24x96-Mode 7 und im 24x132-Mode 5 Bildschirmseiten zwischengespeichert.

jk82 VIDEO I kann bis zu 5 verschiedene Terminals emulieren. 3 Emulationstabellen sind dabei im EPROM vorbesetzt, eine 4. Tabelle kann im EPROM nachprogrammiert werden und eine Tabelle ist vom Computer in die VIDEO I-Platine ladbar.

jk82 VIDEO I kann mit 2 verschiedenen Grafik-Betriebsarten benutzt werden:

- Grafik-Modus mit 768 x 512 Bildpunkten (mit Zeilensprung)
- Grafik-Modus mit 768 x 256 Bildpunkten (ohne Zeilensprung)

In beiden Betriebsarten ist die Grafik-Ansteuerung identisch. Es sind Funktionen zum Setzen, Löschen und Invertieren von Punkten und Vektoren implementiert. Außerdem kann ein vollständiger Alfa-Zeichensatz geladen werden. Die Darstellung von Text und Grafik ist dann auch gemischt möglich.

1.1 Vereinbarungen von Bezeichnungsweisen in diesem Handbuch

Im folgenden soll für die Darstellung von Zeichenfolgen, die zwischen VIDEO I und Computer ausgetauscht werden, folgendes gelten:

<..> bedeutet einen Control-Code (z.B.: <ESC>, <CR>, <LF>)
'..' bedeutet ein darstellbares Zeichen (z.B.: 'X', 'B', 'b')
nnn bedeutet einen Code in Dezimaldarstellung (z.B.: 15, 30)
nnnD bedeutet einen Code in Dezimaldarstellung (z.B.: 15D, 30D)
nnnH bedeutet einen Code in hexadezimaler Darstellung (z.B.: 0AH, 0FFH)
^.. bedeutet ein Control-Zeichen (z.B.: ^I, ^H, ^J)

Kenncode steht für eine Kontrollsequenz, die eine Funktion einleitet. Je nach Terminal-Emulation kann diese Kontrollsequenz verschieden sein. Meist ist jedoch die Sequenz <ESC> 'X' nnn (mit nnn = Funktionsnummer) möglich.

Sowohl im Alfa-Modus als auch im Grafik-Modus können Buchstaben und andere Character dargestellt werden. Das Verhalten der VIDEO I-Platine ist jedoch in diesen beiden Betriebsarten verschieden. Um Verwechselungen zu vermeiden ist in diesem Handbuch folgende Bezeichnungsweise gewählt worden:

ALFA-Modus: Betriebsart zur Darstellung von 'Texten' auf dem Bildschirm mit der Möglichkeit diese Texte mit verschiedenen Attributen zu versehen. In dieser Betriebsart gelten alle Terminal-Emulations-Möglichkeiten wie in Kapitel 2 beschrieben. Punktgrafik ist nicht möglich.

TEXT-Modus: Betriebsart zur Darstellung von 'Texten' ohne Attribute auf dem Bildschirm. Diese Betriebsart kann nur vom GRAFIK-Modus aus selektiert werden. Eine gemischte Darstellung von Texten und Punktgrafik ist in dieser Betriebsart möglich. Terminal-Emulations-Möglichkeiten wie in Kapitel 2 beschrieben existieren hier nicht. Der Text-Modus gehört logisch zur Punktgrafik und ist auch dort beschrieben.

1.2 Gliederung dieses Handbuches

Im Kapitel 2 sind alle vom Computer auslösbare Funktionen nach Gruppen geordnet kurz beschrieben worden. Eine exakte Beschreibung der einzelnen Befehle erfolgt im Kapitel 6. Dort sind alle Funktionen nach Nummern geordnet.

Kapitel 3 beschreibt alle Möglichkeiten des Grafik-Betriebes. Zur Erstellung eines ladbaren Text-Zeichensatzes im Grafik-Betrieb ist die zusätzliche Lektüre des Kapitels 8 zu empfehlen.

Von der Tastatur auslösbare Funktionen sind im Kapitel 4 beschrieben. Falls Änderungen an den Tastatur-Funktionen vorgenommen werden sollen, so ist dies ebenfalls im Kapitel 4 beschrieben.

Kapitel 5 beschreibt die Möglichkeiten der Terminal-Emulationen einschließlich des Erstellens und Ladens von neuen Terminal-Emulationen.

Das Kapitel 7 beschreibt den Zeichensatz, der sich im EPROM der VIDEO I befindet. Falls keine Änderungen an diesem EPROM vorgenommen werden sollen, kann dieses Kapitel getrost überlesen werden.

Die Kapitel 9 und 10 beschreiben zwei interessante Hilfsprogramme der Diskette 'Hilfsprogramme zur VIDEO I'. 'ALFA' ist ein Kommandointerpreter zur Einstellung diverser Funktionen des 'ALFA'-Modus (alle Funktionen des Kapitels 6 können mit 'ALFA' ausgelöst werden). 'GRED' ist ein Grafik-Editor mit dem die Möglichkeiten des Grafik-Modus (einschließlich Hardcopy auf einen Drucker) ausgenutzt werden können.

Im Glossar (Kapitel 12) schließlich werden einige in diesem Handbuch benutzte Begriffe kurz erläutert.

1.3 Hinweise zum Betrieb der VIDEO I als Konsole unter CP/M

Beim Betrieb der VIDEO I als Konsole unter CP/M und dazu kompatiblen Betriebssystemen (z.B. ZDOS) ist zu beachten, daß viele Controlzeichen vom Betriebssystem interpretiert und nicht unverändert an die VIDEO I weitergereicht werden. Das gleiche gilt auch für Zeichen, die von der VIDEO I an den Benutzer gesendet werden sollen (z.B.: ^S). Außerdem werden sehr häufig unerwünschte <CR> und <LF> eingefügt.

Beispiel: Das Zeichen 09H (=TAB) wird bei Ausgabe über das Betriebssystem zu mehreren Blanks expandiert und kann deshalb über das Betriebssystem nicht an die VIDEO I gesendet werden. Das Problem tritt z.B. bei allen Cursor-Funktionen ohne Offset und insbesondere im Grafik-Modus auf.

Bei einigen Funktionen kann deshalb die VIDEO I nicht über einen BDOS-Aufruf (BDOS-Funktionen 1, 2, 6, 9, 10 oder 111) angesprochen werden. Bei Aufruf dieser BDOS-Funktionen ist deshalb folgendes zu beachten:

BDOS-Funktion 1 (console input)

Ein von der VIDEO I übergebenes Byte, das nicht von der Tastatur kommt sondern eine Cursorposition oder ein Byte aus dem Bildspeicher darstellt, kann den ASCII-Code für ^S, ^Q oder ^C annehmen. Solche Kontrollzeichen werden vom Betriebssystem abgefangen und als Kommando interpretiert.

BDOS-Funktion 2 (console output)

Eine Cursor-Koordinate oder eine xy-Koordinate im Grafik-Modus, die den Wert 9 (entspricht ^I) enthält, führt zu einer Tabulator-Expansion, bei der das ^I durch mehrere Leerzeichen (20H) ersetzt wird.
Das Betriebssystem CP/M 3.0 fügt nach einer bestimmten Anzahl von Zeichen ein <cr> <lf> (Wagenrücklauf und Zeilenvorschub) ein.

BDOS-Funktion 6 (direct console input, output)

Ein mit diesem Aufruf ausgegebenes Byte kann den Steuer-Code der Funktion (0ffh, 0feh oder 0fdh) annehmen. Statt einer Ausgabe wird eine Eingabe durchgeführt bzw. lediglich der Status abgefragt.

Für die BDOS-Funktionen 9, 10, 111 gilt ähnliches.

Die obigen Funktionen werden von einem Programm, das in einer höheren Programmiersprache geschrieben ist z.B. durch

```
write (1,101)  x,y          FORTRAN
write (x,y);                PASCAL
print x y                   BASIC
```

aufgerufen. BASIC sendet wie CP/M 3.0 <cr> <lf> nach einer bestimmten Anzahl von Zeichen und expandiert den Tabulator-Befehl.

Falls die Funktion HARDCOPY der VIDEO I verwendet werden soll, darf der Drucker erst über einen BDOS-Aufruf (BDOS-Funktion 5) angesprochen werden, wenn alle angeforderten Bytes (768 pro Aufruf) von der VIDEO I abgeholt wurden. Das Betriebssystem fragt bei

jedem Aufruf die Konsole ab (^S-, ^C-Suche), so daß Bilddaten, die zum Drucker gelangen sollen, verloren gehen.

Um dennoch alle Funktionen nutzen zu können, kann die VIDEO I entweder direkt als I/O-Port oder über einen BIOS-Aufruf angesprochen werden. Für die Form des Aufrufs und die Art der Parameterübergabe sind die Hinweise im Handbuch zum Betriebssystem, zum Compiler (FORTRAN, PASCAL, C, BASIC usw.) oder zum Interpreter (BASIC) zu beachten.

Die genannten Einschränkungen gelten nur in den Sonderfällen, die über den normalen Terminalbetrieb hinausgehen (z.B. bei Grafik oder besonderen Funktionen).

Bei den Terminalemulationen von ADDS Viewpoint oder ADM 3a kann bei der Zeichenausgabe, Cursor-Adressierung, Attributdefinition, Teilscroll usw. mit den BDOS-Aufrufen gearbeitet werden.

In der VISA-Emulation tritt ein Problem bei der Cursor-Positionierung auf, da die Koordinaten nach VISA-Spezifikation ohne Offset übermittelt werden. Um dennoch die Tabulatorexpansion zu umgehen, kann bei der VIDEO I wie auch beim Originalgerät ein Offset von 128 bzw. -128 zu der Cursor-Adresse addiert werden (gesetztes 8. Bit).

1.4 Hinweise zur Benutzung der VIDEO I in Multi-Task-Systemen

Für die Benutzung der VIDEO I in Multi-Task-Systemen bietet der Cursor-Stack eine gute Hilfe zur Aufteilung des Bildschirmes auf die verschiedenen Tasks.

So kann z.B. eine Hintergrund-Task den augenblicklichen Cursor pushen, danach den Cursor neu positionieren, die Nachricht auf den Bildschirm schreiben und dann nach einem 'POP Cursor' die Kontrolle an das Vordergrundprogramm zurückgeben.

Dabei ist jedoch folgendes Problem zu beachten:

Falls die VIDEO I noch auf weitere Bytes einer noch nicht abgeschlossenen ESC-Sequenz wartet, darf kein Task-Switch-Interrupt ausgelöst werden! Falls dies dennoch geschieht, würde die VIDEO I den 'PUSH Cursor'-Befehl nicht erkennen, da die ersten Bytes dieser ESC-Sequenz zuerst für die unterbrochene ESC-Sequenz benutzt würden!

Hier gibt es nur 2 Abhilfemöglichkeiten:

- Sperren von Interrupts bis eine ESC-Sequenz abgearbeitet wurde!
- Alle benutzten Funktionen mit 1-Byte-Control-Codes auslösen! Notfalls eine neue Terminal-Emulation (z.B. Terminal 4 oder 5) erstellen, bei der alle benötigten Funktionen mit Control-Codes ausgelöst werden können! Dies geht allerdings nicht bei Funktionen, die noch weitere Parameter benötigen (z.B. Cursor-Adressierung).

2. Alfanumerischer Modus

Bei Eingabe eines darstellbaren ASCII-Zeichens erfolgt die Wiedergabe auf dem Monitor an die zuvor besetzte Cursorposition. Der Cursor springt dann auf die nächste Stelle. Bei Ende der Zeile wird die Anfangsposition der nächsten Zeile besetzt, ebenso wird bei Ende der Bildschirmposition automatisch ein "scroll up" durchgeführt und der Beginn der nächsten Zeile durch den Cursor belegt.

Bei Eingabe eines nicht darstellbaren ASCII-Zeichens wird die augenblicklich aktive Emulationstabelle interpretiert und die dafür vorgesehene Funktion ausgeführt. Falls das gesendete Zeichen nicht in der Emulationstabelle enthalten ist, erfolgt keine Operation.

VIDEO I ist mit einer großen Anzahl von Steuerfunktionen ausgerüstet, die über Ein-, Zwei- oder Drei-Byte Befehle erreicht werden können. Gemeint ist hiermit der Aufruf einer Funktion und nicht die gesamte Anzahl von Bytes, mit denen die Funktion dargestellt wird. So kann die Steuerfunktion "Direkte Cursoradressierung" durchaus bis zu ihrem Aufruf 3 Byte benötigen (z.B. <ESC> 'A' 'B'), erfordert aber noch zwei weitere Bytes zur Übergabe der X- und Y- Adresse des Cursors.

Zur Zeit sind insgesamt 115 verschiedene Steuerfunktionen implementiert. Die Ansteuerung ist abhängig von der jeweiligen Terminalemulation. Bei einigen Emulationen sind auch nicht alle Funktionen erreichbar. Aus diesem Grunde sind alle Funktionen durchnumeriert worden und können in den Emulationen für Terminal 1 und 2 über die Bytefolge <ESC> 'X' Funktionsnummer erreicht werden.

VIDEO I wird mit einer voreingestellten Belegung der Steuerfunktionen geliefert, die den Tabellen im Anhang zu entnehmen sind. Die genaue Wirkungsweise dieser Funktionen ist im Kapitel 6 beschrieben.

2.1 Cursor- und Bildschirmsteuerung

VIDEO I verfügt über einen sehr leistungsfähigen Satz von Cursor- und Bildschirmsteuerfunktionen. Viele dieser Funktionen sind sich zwar sehr ähnlich, unterscheiden sich aber etwas in ihrer Wirkungsweise. Dies war erforderlich, um die Möglichkeit der Emulation von möglichst vielen Terminals zu schaffen.

Funktionen zur relativen Cursorbewegung:

Funktion 3: **CURSOR ZURÜCK** (backspace)
Funktion 6: **CARRIAGE RETURN**
Funktion 17: **CURSOR NACH UNTEN** (cursor down)
Funktion 18: **CURSOR NACH OBEN** (cursor up)
Funktion 19: **CURSOR LINKS** (cursor left)
Funktion 20: **CURSOR RECHTS** (cursor right)
Funktion 21: **CURSOR VOR** (cursor forward)

Bei den obigen Funktionen wird der Cursor relativ zu seiner augenblicklichen Position auf dem Bildschirm bewegt.

- Zwischen den Funktionen 3 und 19 besteht folgender Unterschied: Falls sich der Cursor bereits am Anfang einer Zeile befindet, verändert die Funktion 19 die Cursorposition nicht, während bei Funktion 3 der Cursor an das Ende der vorhergehenden Zeile springt.

- Zwischen den Funktionen 20 und 21 besteht folgender Unterschied: Falls sich der Cursor bereits am Ende einer Zeile befindet, verändert die Funktion 20 die Cursorposition nicht, während bei Funktion 21 der Cursor an den Anfang der nächsten Zeile springt.

Funktionen zur absoluten Cursorbewegung:

Funktion 9: **ABSOLUTE CURSOR ADRESSIERUNG MIT OFFSET (Y,X)**
(absolute cursor addressing with offset (y,x))
Funktion 10: **ABSOLUTE CURSOR ADRESSIERUNG OHNE OFFSET (Y,X)**
(absolute cursor addressing without offset (y,x))
Funktion 11: **ABSOLUTE CURSOR ADRESSIERUNG MIT OFFSET (X,Y)**
(absolute cursor addressing with offset (x,y))
Funktion 12: **ABSOLUTE CURSOR ADRESSIERUNG OHNE OFFSET (X,Y)**
(absolute cursor addressing without offset (x,y))
Funktion 36: **CURSOR HOME**

Bei den obigen Funktionen wird der Cursor unabhängig von seiner vorherigen Position auf dem Bildschirm positioniert. 'Y' gibt dabei die Zeilennummer und 'X' die Spaltennummer an. Gezählt wird dabei von der oberen linken Ecke, beginnend mit Zeile 0 und Spalte 0. Die 'HOME-Position' des Cursors befindet sich in Zeile 0 und Spalte 0. Die Funktionen unterscheiden sich in der Reihenfolge der Übertragung von Zeile und Spalte sowie in der Art der Beschreibung von Zeile und Spalte (mit oder ohne Offset). Bei Übertragung mit Offset wird dieser mit 32 = 20H (ASCII-BLANK) erwartet.

Die Bereichsgrenzen für Zeile und Spalte werden nicht überprüft! Bei einer Cursorpositionierung über die Zeilen- und Spaltengrenzen des Bildschirmes hinaus ist das Ergebnis nicht vorhersehbar!

Funktionen zur Positionierung des Cursors innerhalb einer Zeile bzw. innerhalb einer Spalte:

Funktion 13: **HORIZONTALE CURSOR ADRESSIERUNG**
(horizontal cursor addressing)
Funktion 14: **VERTIKALE CURSOR ADRESSIERUNG**
(vertical cursor addressing)

Die Cursorposition wird bei Funktion 13 innerhalb einer Zeile verändert. Bei Funktion 14 wird die Cursorposition innerhalb einer Spalte verändert.

Die Zeilen- bzw. Spaltenposition wird immer mit einem Offset von 32 = 20H (ASCII-BLANK) erwartet.

Die Bereichsgrenzen für Zeile und Spalte werden auch hier nicht überprüft! Bei einer Cursorpositionierung über die Zeilen- und Spaltengrenzen des Bildschirmes hinaus ist das Ergebnis nicht vorhersehbar!

Cursor-Stack-Funktionen

Funktion 112: **PUSH CURSOR**
Funktion 113: **POP CURSOR**
Funktion 114: **SWAP CURSOR**
Funktion 115: **RESET CURSOR STACK**

jk82 VIDEO I stellt intern einen Stack für 1000 Cursor-Positionen zur Verfügung. Auf diesem Stack sind die Funktionen 112 bis 115 wie folgt definiert:

Wenn HL ein Register für Cursor-Positionen und SP ein Zeiger auf den Cursor-Stack ist, dann sind die obigen Funktionen analog zu den folgenden Z80-Funktionen zu verstehen:

```
Funktion 112: push  hl
Funktion 113: pop   hl
Funktion 114: ex    (sp),hl
Funktion 115: ld    sp,stack_top
```

Es ist vom Anwender sicherzustellen, daß kein Stack-Überlauf eintritt.

Funktion zum Zurücksenden eines Zeichens an den Computer:

Funktion 22: **ZEICHEN AN CURSOR-POSITION SENDEN**
(send char at cursor)

Funktionen zur Übermittlung der augenblicklichen Cursorposition an den Computer:

Funktion 23: **CURSOR-ADRESSE MIT OFFSET SENDEN (Y,X)**
(send cursor address with offset (y,x))
Funktion 24: **CURSOR-ADRESSE OHNE OFFSET SENDEN (Y,X)**
(send cursor address without offset (y,x))
Funktion 25: **CURSOR-ADRESSE MIT OFFSET SENDEN (X,Y)**
(send cursor address with offset (x,y))
Funktion 26: **CURSOR-ADRESSE OHNE OFFSET SENDEN (X,Y)**
(send cursor address without offset (x,y))

Die Übermittlung der augenblicklichen Cursorposition an den Computer erfolgt analog zu den Cursorpositionierungsfunktionen entweder in der Reihenfolge Zeile/Spalte oder Spalte/Zeile. Auch wird hier zwischen Funktionen mit Offset und solchen ohne Offset unterschieden.

Die beiden vorhergehenden Funktionsgruppen senden eine Zeichenfolge an den Computer zurück. Diese kann vom Computer in der gleichen Art gelesen werden, wie Eingaben von der an der VIDEO I angeschlossenen Tastatur. Daraus ergibt sich aber auch gleich das Problem, daß diese Zeichenfolgen nicht von Tastatureingaben unterschieden werden können! Zufällig noch nicht ausgelesene Tastatureingaben führen zwangsläufig zu falschen Werten!

Es wird daher folgende Vorgehensweise empfohlen (in der angegebenen Reihenfolge vorgehen):

1. Tastatur blockieren (Funktion 83)
2. VIDEO I Datenport auslesen (ohne Statusabfrage)
3. Zeichenfolge von der VIDEO I anfordern (Funktion 22 bis 26)
4. Die angeforderten Zeichen auslesen (mit Statusabfrage)
5. Tastaturblockade aufheben (Funktion 84)

Funktionen zur Bildverschiebung (scroll):

Funktion 4: **ZEILENVORSCHUB** (linefeed)
Funktion 5: **UMGEKEHRTER ZEILENVORSCHUB** (reverse linefeed)
Funktion 7: **CARRIAGE RETURN + LINEFEED**
Funktion 27: **TEILSCROLL ABWAERTS** (partial scroll down)
bzw. **ZEILE EINFUEGEN** (insert line)
Funktion 28: **TEILSCROLL AUFWAERTS** (partial scroll up)
Funktion 29: **SCROLL NACH UNTEN** (scroll down)
Funktion 30: **SCROLL NACH OBEN** (scroll up)
Funktion 95: **FENSTER-SCROLL NACH UNTEN** (window scroll down)
Funktion 96: **FENSTER-SCROLL NACH OBEN** (window scroll up)

Bei den Funktionen 4,5 und 7 erfolgt eine Bildverschiebung nur bei Bedarf. Falls sich der Cursor bei Funktion 5 nicht in der Zeile 0 befand, wirkt Funktion 5 genauso wie Funktion 18 (Cursor nach oben). Falls sich der Cursor bei Funktion 4 nicht in der letzten Bildschirmzeile befand, wirkt Funktion 4 genau so wie Funktion 17 (Cursor nach unten).

Die Funktionen 95 und 96 sind unabhängig von der augenblicklichen Cursorposition. Die Koordinaten des linken oberen Eckpunktes und des rechten unteren Eckpunktes müssen jeweils mit übertragen werden.

Funktionen zum Wechseln von Bildschirmseiten:

Funktion 31: **AKTUELLE BILDSCHIRMSEITE EINSTELLEN** (page actual)
Funktion 32: **SEITE ZURÜCK** (page down)
Funktion 33: **SEITE VOR** (page up)

Mit den obigen Funktionen kann zwischen den verschiedenen Bildschirmseiten der VIDEO I geblättert werden. Diese Funktionen können auch direkt von der Tastatur ausgelöst werden (s. Kapitel 4).

Funktionen zur Löschung von Bildschirminhalten:

Funktion 8: **CARRIAGE RETURN UND REST DER ZEILE LÖSCHEN**
(carriage return and erase end of line)
Funktion 34: **ZEILE LÖSCHEN** (erase line)
Funktion 35: **REST DER ZEILE LÖSCHEN** (erase end of line)
Funktion 37: **BILDSCHIRM LÖSCHEN UND CURSOR HOME**
(erase screen and cursor home)
Funktion 38: **BILDSCHIRM LÖSCHEN** (erase screen)
Funktion 39: **ANFANG DES BILDSCHIRMS LÖSCHEN** (erase start of screen)
Funktion 40: **ENDE DES BILDSCHIRMS LÖSCHEN** (erase end of screen)
Funktion 57: **SEITENVORSCHUB MIT LÖSCHEN DES BILDSCHIRMES**
(formfeed and clear screen)
Funktion 97: **FENSTER-INHALT LÖSCHEN** (clear window)

Bei den Funktionen 37, 38 und 57 wird der gesamte Bildschirm gelöscht, während bei den anderen Funktionen nur Teile des Bildschirms gelöscht werden. Mit Ausnahme der Funktion 97 sind diese Teile jeweils abhängig von der augenblicklichen Cursorposition.

Bei Funktion 57 wird vor dem Löschen des Bildschirmes intern eine Seite nach vorn 'geblättert', so daß der alte Bildschirminhalt durch 'Zurückblättern' wieder erreichbar ist.

Funktion 97 löscht ein Bildschirmfenster, dessen Eck-Koordinaten zusätzlich übertragen werden müssen.

Funktionen zum Ein- und Ausschalten des Cursors:

Funktion 85: **CURSOR AUS** (cursor off)
Funktion 86: **CURSOR BLINKEN EIN** (cursor blink on)
Funktion 87: **CURSOR EIN** (cursor on)

Der Cursor wird sowohl bei Funktion 86 als auch bei Funktion 87 eingeschaltet. Bei Funktion 86 erscheint der Cursor als invers blinkender Block auf dem Bildschirm und bei Funktion 87 als nicht blinkender Block.

Funktionen zum Ein- und Ausschalten der Bildschirmanzeige

Funktion 98: **ZEICHENDARSTELLUNG AUF DEM BILDSCHIRM EINSCHALTEN**
(screen dark off)
Funktion 99: **ZEICHENDARSTELLUNG AUF DEM BILDSCHIRM AUSSCHALTEN**
(screen dark on)

Mit Hilfe der obigen Funktionen kann die Anzeige auf dem Bildschirm ein- und ausgeschaltet werden, ohne daß Löschaktionen im Bildwiederholspeicher stattfinden.

Nach Auslösen der Funktion 99 findet zwar keine Bildschirmanzeige mehr statt, aber alle VIDEO I-Funktionen werden intern ohne irgendwelche Beeinträchtigungen ausgeführt.

Diese Funktionen können z.B. genutzt werden, um die Bildröhre eines angeschlossenen Monitors zu schonen, wenn keine Aktionen auf einem ansonsten bereiten Computer stattfinden (gilt nicht für inverse Darstellung).

In der Revision 1.0 der VIDEO I-Hardware ist das Ausschalten der Zeichendarstellung auf dem Bildschirm nicht möglich. Die Funktion 99 bleibt deshalb dort ohne Wirkung.

2.2 Attribute

Mit VIDEO I sind 4 Attribute für die Darstellung einzelner Zeichen möglich:

I. Invers
II. Blinken
III. Unterstreichen
IV. Halbe Helligkeit

Alle Attribute können beliebig kombiniert werden, womit sich 16 unterschiedliche Darstellungsarten ergeben. Sie haben die Funktion einer Voreinstellung, d.h., bei Aufruf eines Attributes werden alle nachfolgenden Zeichen mit diesem voreingestellten Attribut dargestellt, bis dieses wieder abgeschaltet oder ein anderes Attribut gewählt wird. Folgende Funktionen sind direkt aufrufbar:

Funktion 80: **CHARACTER INVERS EIN** (character invers on)
Funktion 79: **CHARACTER INVERS AUS** (character invers off)
Funktion 78: **CHARACTER BLINKEN EIN** (blink on)
Funktion 77: **CHARACTER BLINKEN AUS** (blink off)
Funktion 76: **CHARACTER UNTERSTREICHEN EIN** (underline on)
Funktion 75: **CHARACTER UNTERSTREICHEN AUS** (underline off)
Funktion 74: **CHARACTER HALBE HELLIGKEIT EIN** (half video on)
Funktion 73: **CHARACTER HALBE HELLIGKEIT AUS** (half video off)

Mit den Funktionen 73 bis 80 lassen sich alle 16 Kombinationsmöglichkeiten einstellen, da das aufgerufene Attribut zum voreingestellten, bisher benutzten hinzugefügt wird.

Im Gegensatz hierzu stellen die nachstehenden Funktionen ausschließlich die genannten Attribute ein, d.h., eventuelle andere bereits vereinbarte Attribute werden gelöscht.

Funktion 72: **CHARACTER NORMAL** (normal video)
Funktion 71: **CHARACTER NUR INVERS** (single invers only)
Funktion 70: **CHARACTER NUR BLINKEN** (blink only)
Funktion 69: **CHARACTER NUR UNTERSTREICHEN** (underline only)
Funktion 68: **CHARACTER NUR HALBE HELLIGKEIT** (half intensity only)
Funktion 67: **CHARACTER HALBE HELLIGKEIT UND BLINKEN**
(half intensity blink)
Funktion 66: **CHARACTER HALBE HELLIGKEIT UND INVERS**
(single invers + half intensity)
Funktion 65: **CHARACTER BLINKEN UND INVERS**
(single invers + blink)
Funktion 64: **CHARACTER HALBE HELLIGKEIT, BLINKEN UND INVERS**
(single invers + half intensity + blink)
Funktion 63: **CHARACTER HALBE HELLIGKEIT UND UNTERSTREICHEN**
(half intensity + underline)
Funktion 62: **CHARACTER BLINKEN UND UNTERSTREICHEN**
(underline + blink)
Funktion 61: **CHARACTER HALBE HELLIGKEIT, BLINKEN UND UNTERSTREICHEN**
(underline + half intensity + blink)

'Unterstreichen' wird bei Benutzung eines anderen geeigneten Charactergenerators als Befehl zum Umschalten auf einen anderen Zeichensatz interpretiert.

Das Attribut 'Unterstreichen' steht bei Bildformaten mit 12 Zeilen pro Bildschirmseite und bei Bildformaten mit weniger als 80 Zeichen pro Zeile nicht zur Verfügung. Falls bei einem solchen Format 'underline' selektiert wird, so bleibt dies ohne Wirkung.

2.3 Tastaturbeeinflussung

VIDEO I verfügt über einen Satz von Funktionen zur Einstellung von diversen Tastatureigenschaften. Diese einmal eingestellten Eigenschaften bleiben auch im Grafik-Modus erhalten. Eine Definition ist jedoch nur im Alfa-Modus möglich, da nur im Alfa-Modus die Emulationstabellen interpretiert werden.

Funktionen zum Ein- und Ausschalten der Tastatur:

Funktion 83: **TASTATUR AUSSCHALTEN** (keyboard lock)
Funktion 84: **TASTATUR EINSCHALTEN** (keyboard unlock)

Bei ausgeschalteter Tastatur werden keine Eingaben von der Tastatur mehr angenommen, bis die Tastatur wieder eingeschaltet wird. Die Funktion 'TASTATUR EINSCHALTEN' kann jedoch auch dann mit der gleichen Code-Folge (wie vom Computer) auch von der Tastatur ausgelöst werden. Die dafür gültige Code-Folge ist abhängig von der jeweils gültigen Terminal-Emulation.

Definition von Zeichenfolgen auf Funktionstasten und Ändern der Tastencode-Belegung:

Funktion 56: **TASTATURCODE ÄNDERN**
(change character code set -- keyboard)
Funktion 58: **ZEICHENFOLGEN LÖSCHEN** (clear string buffer)
Funktion 59: **ZEICHENFOLGEN LADEN** (load string)

Tasten, denen ein Code mit gesetztem 8. Bit zugeordnet ist, können mit einem beliebigen String belegt werden, der immer dann an den Computer gesendet wird, wenn diese Taste gedrückt wird. Dabei ist aber zu beachten, daß nur ein begrenzter Platz für den Stringpuffer (2KByte) zur Verfügung steht. Bei Überschreitung dieser Grenze werden keine weiteren Zeichenfolgen mehr angenommen. Obwohl der Zeichenpuffer mit 2KByte recht groß gewählt ist, kann ein Überlauf dann eintreten, wenn das Laden von Zeichenfolgen öfters aufgerufen wird, ohne vorher den Puffer zu löschen.

Mit Funktion 56 können alle Tastencodes (auch ohne gesetztes 8. Bit) 1:1 in einen anderen Code umdefiniert werden. Die Funktionsweise ist analog zur Funktion 55.

Funktionen zur Beeinflussung des Tastaturpuffers:

Funktion 89: **TASTATUR-PUFFER DEAKTIVIEREN**
(keyboard buffer not activ)
Funktion 90: **TASTATUR-PUFFER AKTIVIEREN**
(keyboard buffer activ)
Funktion 91: **TASTATUR-PUFFER LÖSCHEN**
(clear keyboard buffer)

VIDEO I speichert alle von der Tastatur eingegebenen Zeichen in einem 256 Byte großen Zeichenpuffer und gibt sie aus diesem anschließend bei Bedarf an den Computer weiter. Dies hat den Vorteil, daß bereits Eingaben 'auf Vorrat' möglich sind. In einigen Fällen ist es jedoch wünschenswert, den Inhalt des Tastaturpuffers zu löschen oder gar die Zwischenpufferung von Zeichen zu unterbinden.

Bei deaktiviertem Tastaturpuffer wird jedes eingegebene Zeichen von der VIDEO I sofort an die Computerschnittstelle weitergegeben. Falls der Computer das vorhergehende Zeichen noch nicht ausgelesen hat, geht es verloren.

Strings werden auch bei deaktiviertem Tastaturpuffer expandiert.

2.4 Bildschirmformate einstellen

Funktion 50: **24 ZEILEN MIT 132 ZEICHEN PRO ZEILE**
(132 characters per line, 24 lines per screen)
Funktion 51: **24 ZEILEN MIT 96 ZEICHEN PRO ZEILE**
(96 characters per line, 24 lines per screen)
Funktion 52: **24 ZEILEN MIT 80 ZEICHEN PRO ZEILE**
(80 characters per line, 24 lines per screen)
Funktion 100: **25 ZEILEN MIT 132 ZEICHEN PRO ZEILE**
(132 characters per line, 25 lines per screen)
Funktion 101: **25 ZEILEN MIT 96 ZEICHEN PRO ZEILE**
(96 characters per line, 25 lines per screen)
Funktion 102: **25 ZEILEN MIT 80 ZEICHEN PRO ZEILE**
(80 characters per line, 25 lines per screen)
Funktion 103: **24 ZEILEN MIT 66 ZEICHEN PRO ZEILE**
(66 characters per line, 24 lines per screen)
Funktion 104: **24 ZEILEN MIT 48 ZEICHEN PRO ZEILE**
(48 characters per line, 24 lines per screen)
Funktion 105: **24 ZEILEN MIT 40 ZEICHEN PRO ZEILE**
(40 characters per line, 24 lines per screen)
Funktion 106: **12 ZEILEN MIT 132 ZEICHEN PRO ZEILE**
(132 characters per line, 12 lines per screen)
Funktion 107: **12 ZEILEN MIT 96 ZEICHEN PRO ZEILE**
(96 characters per line, 12 lines per screen)
Funktion 108: **12 ZEILEN MIT 80 ZEICHEN PRO ZEILE**
(80 characters per line, 12 lines per screen)
Funktion 109: **12 ZEILEN MIT 66 ZEICHEN PRO ZEILE**
(66 characters per line, 12 lines per screen)
Funktion 110: **12 ZEILEN MIT 48 ZEICHEN PRO ZEILE**
(48 characters per line, 12 lines per screen)
Funktion 111: **12 ZEILEN MIT 40 ZEICHEN PRO ZEILE**
(40 characters per line, 12 lines per screen)

Nach Auslösung einer der obigen Funktionen wird der Bildschirm sofort in die neue Darstellungsart umgeschaltet. Gleichzeitig wird der Bildschirm gelöscht und der Cursor wird in Zeile 0 und Spalte 0 (home) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Die Funktionen 103 bis 111 stehen nicht in allen Versionen der VIDEO I zur Verfügung!

2.5 Zeichensätze anwählen

Im Zeichengenerator der VIDEO I-Platine befinden sich 2 vollständige Zeichensätze. Außerdem sind auch die Control-Codes in beiden Character-Sätzen mit druckbaren Zeichen belegt worden. Der 2. Zeichensatz beinhaltet die Zeichen des 1. Satzes in unterstrichener Form und wird durch die Funktion 'UNDERLINE' (siehe 'Attribute') angewählt. Auf den Control-Codes des 2. Zeichensatzes sind jedoch zum Teil andere Zeichen vereinbart worden. Die vollständige Belegung der beiden Zeichensätze kann dem Anhang entnommen werden. Dabei ist jedoch zu beachten, daß die Zeichen 80H bis 0FFH in den Tabellen des Anhanges zum jeweils 2. Zeichensatz gehören und nur durch Aktivieren des 2. Zeichensatzes (z.B. durch Funktion 76)

erreicht werden können! Insbesondere können diese Zeichen nicht durch Setzen des 8. Bites erreicht werden!

Der 2. Zeichensatz ist nur bei den Bildformaten 24 oder 25 Zeilen mit 80, 96 oder 132 Zeichen pro Zeile vorhanden. Bei allen anderen Bildformaten steht ein 2. Zeichensatz (und damit normalerweise das Attribut 'underline') nicht zur Verfügung.

Die Zeichen mit den Codes 00H bis 1FH können nicht direkt angewählt werden, da VIDEO I bei diesen Codes die zugehörige Controlfunktion ausführt. Diese Zeichen können jedoch ganz einfach auf 2 verschiedene Arten zur Anzeige gebracht werden:

1. Die Codes 00H bis 1FH werden mit einem gesetzten 8. Bit als Codes 80H bis 9FH an die VIDEO I übertragen. Da auch bei den anderen darzustellenden Codes (20H bis 7FH) das 8. Bit gesetzt sein darf, kann grundsätzlich bei allen darzustellenden Zeichen das 8. Bit gesetzt werden. Nur bei Controlzeichen, die tatsächlich die zugehörige Controlfunktion auslösen sollen, darf das 8. Bit nicht gesetzt sein.

2. Falls der verwendete Computer die Übertragung von Zeichen mit gesetztem 8. Bit nicht zuläßt oder aber eine Codegleichheit mit anderen Geräten hergestellt werden soll, besteht noch die Möglichkeit Zeichen mit beliebigem Code auf Zeichen mit einem beliebigen anderen Code abzubilden. Auf diese Art ist es möglich, z.B. die Zeichen mit den Codes 00H bis 1FH auf darstellbare Zeichen abzubilden. Dies ist mit den nachfolgenden 3 Funktionen durchführbar.

Funktion 53: **US - ZEICHENSATZ** (US character set)
Funktion 54: **DEUTSCHER ZEICHENSATZ** (german character set)
Funktion 55: **CODE ÄNDERN** (change character code set)

Der deutsche Zeichensatz ist im Charactergenerator im Controlcode-Teil enthalten. Nach Ausführung der Funktion 54 werden diese Controlcodes auf die korrekten, sonst mit den amerikanischen Zeichen belegten, Codes abgebildet. Mit Funktion 53 wird diese Abbildung wieder zurückgenommen. Da die zugehörigen Zeichen im Bildwiederholspeicher jedoch verschieden sind, ist eine gleichzeitige Darstellung beider Zeichensätze auf dem Bildschirm möglich. Die Funktion 53 und 54 müssen dann wechselseitig ausgeführt werden.

Mit Funktion 55 kann die Abbildung eines einzelnen Zeichens auf ein anderes vorgenommen werden.

Beispiel für die Anwendung der Funktion 55:

Bei einigen Typenraddruckern sind die Zeichen '<' und '>' nicht vorhanden. Statt dessen stehen auf diesem Code die Zeichen 'µ' und '°' zur Verfügung. Diese beiden Zeichen sind im Zeichensatz der VIDEO I mit den Codes 0AH und 0EH enthalten. Mit Hilfe der Funktion 55 können nun auch auf dem Bildschirm die Zeichen '<' und '>' durch die Zeichen 'µ' und '°' ausgetauscht werden. Zu diesem Zwecke ist folgende Codefolge an die VIDEO I zu senden:

<ESC> 'X' 55 '<' 0AH <ESC> 'X' 55 '>' 0EH

2.6 Sonstige Funktionen

Funktion 41: **RESET VIDEO I**
Funktion 60: **IDENTIFIKATION**
Funktion 82: **INVERSER BILDSCHIRM EIN**
(invers screen on)
Funktion 81: **INVERSER BILDSCHIRM AUS**
(invers screen off)
Funktion 88: **KLINGEL** (bell)
Funktion 92: **TEST-BILD AUSGEBEN** (display test pattern)
Funktion 93: **NÄCHSTES ZEICHEN IGNORIEREN** (ignore next char)
Funktion 94: **DIREKTES LADEN EINES 6845-REGISTERS**
(load 6845 direct)

Einzelheiten zu den obigen Funktionen sind im Kapitel 6 beschrieben.

3. Grafik-Modus

VIDEO I kennt 2 verschiedene Grafik-Betriebsarten, die sich in der Punktauflösung unterscheiden. Die Anwahl dieser Grafik-Betriebsarten muß vom Alfa-Modus aus durch Aufruf einer Steuerfunktion erfolgen. Soll die Darstellung invertiert erfolgen, d.h. heller Hintergrund, ist vor Einschalten des Grafik-Modus der Befehl "inverser Bildschirm ein" (Funktion 82) aufzurufen.

In beiden Grafik-Modi ist es möglich Buchstaben und Ziffern mit Zeichnungen zu kombinieren. Zu diesem Zweck kann der Text-Modus selektiert werden. Dieser ist eine Unterbetriebsart des Grafik-Modus.

3.1 Grafik-Betriebsart einstellen

Die Grafik-Betriebsart muß vom Alfa-Modus aus angewählt werden. Dazu existieren im Alfa-Modus zwei Funktionen:

Funktion 43: **GRAFIK MODE 1** (grafic mode 1)

VIDEO I wird in den Grafik-Modus mit 768 x 512 Bildpunkten geschaltet. In dieser Betriebsart arbeitet VIDEO I mit dem Zeilensprungverfahren (interlace). Die Vollbildfrequenz beträgt dabei allerdings nur noch 25Hz.

Es ist zu beachten, daß für den GRAFIK MODE 1 mit Zeilensprungverfahren ein Monitor mit hoher Nachleuchtdauer verwendet werden sollte, da sich sonst ein störendes Flackern besonders bei Darstellung horizontaler Linien bemerkbar macht.

Funktion 42: **GRAFIK MODE 2** (grafic mode 2)

VIDEO I wird in den Grafik-Modus mit 768 x 256 Bildpunkten geschaltet. Die logische Ansteuerung erfolgt jedoch genau wie im Mode 1 mit 768 x 512 Bildpunkten. Die Umrechnung auf 768 x 256 Bildpunkte wird von der VIDEO I intern vorgenommen. In dieser Betriebsart wird kein Zeilensprungverfahren (non-interlace) angewendet, so daß die Bildfrequenz 50Hz beträgt.

3.2 Darstellung von alfanumerischen Zeichen im Text-Modus

Im Text-Modus kann die VIDEO I in der Standardversion 4 ladbare Zeichensätze verwalten. In den Versionen 3, 4 und 5 der VIDEO I-Firmware kommen noch zwei weitere feste Zeichensätze im EPROM dazu.

Nach dem Umschalten vom Alfa-Modus in den Grafik-Modus 1 wird Zeichensatz 5 und nach Umschalten in den Grafik-Modus 2 Zeichensatz 6 voreingestellt. Diese beiden Zeichensätze befinden sich bei den Versionen 3, 4 und 5 im EPROM.

In der Standardversion der VIDEO I ist Zeichensatz 5 identisch mit Zeichensatz 1 und Zeichensatz 6 steht für Zeichensatz 2.

Die Voreinstellung kann vom Grafik-Modus nur durch das Umschalten mit <CR> in den Text-Modus erreicht werden. Im Normalfall erfolgt das Umschalten direkt durch die Auswahl des Zeichensatzes (Befehle 'C', 'c', 'D', 'd', 'F' oder 'f').

In der Standardversion der VIDEO I ist die Darstellung von Buchstaben und Ziffern im Grafik-Modus (Text-Modus) nur nach vorherigem Laden eines Zeichensatzes möglich.

Die VIDEO I besitzt die Speicherkapazität für 4 Zeichensätze. Jeweils zwei Zeichensätze belegen einen durchgehenden Speicherbereich von 4096 Byte.

Die 4096 Byte teilen sich auf nach:

2816 Byte für Zeichensatz 1 (bzw. 3)
1280 Byte für Zeichensatz 2 (bzw. 4).

Jedes darstellbare Zeichen setzt sich aus einer Folge von n Byte zusammen. Jedes Byte besitzt m gültige Bit. Ein Zeichen wird demnach durch eine n mal m Punktmatrix dargestellt. Für die Parameter n und m gelten die Grenzwerte:

Zeichenhöhe	n <= 255
Zeichenbreite	m <= 8.

Im Text-Modus unterscheidet die VIDEO I 128 darstellbare Zeichen. Es handelt sich dabei um die 96 Zeichen des ASCII-Satzes erweitert um die Buchstaben des deutschen Zeichensatzes und einigen Sonderzeichen. Adressiert werden die Character durch einen 8-Bit-Code nach ASCII. Außer bei den Kontrollzeichen (00h ... 01fh) ist das achte Bit ohne Bedeutung und wird in der VIDEO I ignoriert. Kontrollzeichen mit gesetztem achten Bit (80h ... 8fh) lösen nicht die ihnen zugeordnete Funktion aus, sondern sprechen eines der Sonderzeichen des Zeichensatzes an und bilden es auf dem Bildschirm ab.

Das Bild des ersten Zeichens (anwählbar über 80h) enthält zwei Byte für die Beschreibung der Struktur des Zeichensatzes, und geht so als darstellbares Zeichen verloren.

Die zu ladende Datenmenge für einen Zeichensatz umfaßt n mal 128 Byte. n ist so zu wählen, daß der verfügbare Speicherraum nicht überschritten wird. Der erste Zeichensatz eines 4096Byte-Blockes kann bei Bedarf den zweiten, der dann nicht selektiert werden darf, überschreiben.

Als Beispiel für die Struktur des 4096Byte-Blockes seien hier Zeichensatz 5 und 6 beschrieben, die in der erweiterten Version der VIDEO I enthalten sind.

Zeichensatz 5 belegt 2560 Byte. Jedes Zeichen wird durch 20 Byte dargestellt; jedes Byte enthält 8 gültige Bit, d.h.

n = 20,
m = 8.

Die Zahlen n und m belegen die ersten beiden Byte des ersten Zeichens. Dieses Zeichen kann zwar, sollte jedoch nicht auf dem Bildschirm dargestellt werden.

256 Byte hinter Zeichensatz 5 sind nicht belegt.

Zeichensatz 6 belegt die restlichen 1280 Byte des 4096Byte-Blockes. Jedes Zeichen wird mit 10 Byte dargestellt. Die Anzahl der gültigen Bit pro Byte beträgt 8. Es gilt:

n = 10
m = 8

Die Steuerbyte n und m sind wieder im ersten Zeichen eingelagert.

Die Zeichen des Zeichensatzes 5 sind doppelt so hoch wie die Zeichen des Zeichensatzes 6. Auf dem Bildschirm dargestellt ergeben sich gleiche Buchstabenhöhen, wenn im Grafik-Modus 1 der Zeichensatz 5 und im Grafik-Modus 2 der Zeichensatz 6 verwendet wird.

3.3 Grafik-Befehle

VIDEO I kennt im Grafik-Modus Befehle zum Manipulieren einzelner Punkte, Punktreihen (Linien) und Punktgruppen zu 8 Bit. Die Ansteuerung ist für beide Grafik-Betriebsarten in den meisten Fällen gleich. Im Text-Modus können zusätzlich ASCII-Zeichen dargestellt werden.

Außerdem besteht die Möglichkeit zum Auslesen einzelner Punkte, Punktgruppen zu 8 Bit und ganzer Punktreihen.

Einige Grafik-Befehle sind im Text-Modus nicht möglich, andere nur im Text-Modus ausführbar oder haben dort eine andere Bedeutung.

Beispiel:

- Das Setzen von Punkten ist im Text-Modus nicht möglich.
- Das Löschen des Bildschirms ist in beiden Modi mit ^L möglich.
- Der Befehl, der mit dem Zeichen 'V' im Grafik-Modus angesprochen wird (Vektor zeichnen), bringt im Text-Modus den Buchstaben 'V' auf den Bildschirm.

3.3.1 Arten der Bildpunktmanipulation

Mit den Punkt- und Vektorbefehlen können Bildpunkte in drei Arten verändert werden:

- setzen
- löschen
- invertieren.

Die Art wird einmal eingestellt und gilt für alle nachfolgenden Punkt- oder Vektorbefehle bis zur nächsten Änderung.

Mit dem Byte-Setz-Befehl können Bitgruppen zu 8 Bit manipuliert werden. Dabei ist es unerheblich, ob eine Bitgruppe zu 8 Bit tatsächlich einem physikalischen Byte in der VIDEO I entspricht oder nicht.

Im Text-Modus kann zusätzlich noch ein Text über die Grafik gelegt werden.

3.3.2 Übergabe der X-Y-Koordinaten

Zur Anwahl eines Bildpunktes ist die Übergabe seiner X-Y-Koordinaten erforderlich. Die Lage des Koordinatensystems geht aus der unten aufgeführten schematischen Bildschirmdarstellung und Kennzeichnung der Eckpunktwerte hervor.

Eck-Koordinaten, 768 x 512 Bildpunkte
(Bei 768 x 256 Bildpunkten wird Y in der Video-Platine durch 2 dividiert, so daß sich die gleichen logischen Koordinaten ergeben.)

X- und Y-Wert müssen als binär codierte Dualzahl byteweise mit vorangestelltem ESC-Steuerzeichen übergeben werden.
Die Reihenfolge der vom Computer an die VIDEO I zu sendenden Werte für eine Koordinate (Punktsequenz) ergibt sich folgendermaßen:

1. Byte: ESC = 1BH (^Ä); Steuerzeichen zur Kennzeichnung der Koordinatenübergabe
2. Byte: low X - niederwertiges Byte des 16 Bit X-Wertes
3. Byte: high X - höherwertiges Byte des 16 Bit X-Wertes
4. Byte: low Y - niederwertiges Byte des 16 Bit Y-Wertes
5. Byte: high Y - höherwertiges Byte des 16 Bit Y-Wertes

Eine vollständige Punktsequenz besteht immer aus 5 Bytes!

Achtung: In der Grafik-Betriebsart 2 werden alle an die VIDEO I übertragenen Y-Koordinaten durch 2 dividiert! Dadurch ergeben sich für den Anwender in beiden Betriebsarten die gleichen anzusteuernden Koordinaten. Beim Wiederauslesen von Bytes oder Zeilengruppen aus der VIDEO I werden jedoch nur die tatsächlich dargestellten Zeilen berücksichtigt.
Beispiel: Der Anwender möchte im Grafik-Betrieb 2 ein Byte in Zeile 5 (in der auf dem Bildschirm dargestellten Zeile 5!) auslesen. Dazu muß für die Grafik-Koordinate vom Anwender die Zeile 10 angewählt werden, da die VIDEO I im Grafik-Modus 2 alle Y-Koordinaten durch 2 dividiert! Falls der Anwender die Zeile 5 anwählt, wird der Grafik-Cursor in die Zeile 2 positioniert. Bei Anwahl der Zeile 4 ebenso.

3.3.3 Befehlsübersicht

3.3.3.1 Im Text- und normalen Grafik-Modus gültige Befehle

Bildschirm	löschen	^L = <FF>
Bildschirm löschen und Grafik-Modus einschalten		^Y = <EM>
Zeichensatz	laden	^B = <STX> oder ^C = <ETX>

3.3.3.2 Nicht im Text-Modus gültige Befehle

Punktmanipulation	setzen	'+'
Punktmanipulation	löschen	'-'
Punktmanipulation	invertieren	'*'
Punkt	ändern	punktsequenz
Koordinate	setzen	'K'
Punkt	holen	'G'
Vektor	zeichnen	'V'
Vektor von c.p.	zeichnen	'v'
Byte	setzen	'S'
Byte	holen	'R'
Hardcopy		'H'
Text-Modus	einschalten	<CR>,'C','c','D','d','F' oder 'f'
Bildschirm	löschen	^J = <LF>
Bildschrim	dunkel steuern	^Q = <DC1>
Bildschirm	hell steuern	^P = <DLE>
Grafik-Modus	verlassen	'E'

3.3.3.3 Nur im Text-Modus gültige Befehle

Zeichen	setzen	' ' ... 'ß', DEL
Cursor	rechts	^I = <HT>
Cursor	links	^H = <BS>
Cursor	hoch	^K = <VT>
Cursor	tief	^J = <LF>
Cursor an Zeilenanfang		^M = <CR>
Textmodus verlassen		^Ö = <FS> oder ^Ü = <GS>

3.3.4 Beschreibung der Befehle

3.3.4.1 Im Text- und normalen Grafik-Modus gültige Befehle

Bildschirm löschen: ^L = <FF>

Der Befehl löscht den Bildschirm.

Bildschirm löschen: ^Y = <EM>

Der Befehl löscht den Bildschirm und schaltet die Voreinstellung für Punkt setzen ('+') ein. Im Textmodus schaltet dieser Befehl gleichzeitig in den Grafik-Modus zurück.

Zeichensatz laden: ^B=<STX>, 4096 Zeichen (Zeichensätze 1 und 2)

Der Befehl besetzt den Zeichengenerator mit den Zeichen, die im Text-Modus auf den Bildschirm gelangen sollen. Dem Steuerzeichen (^B) folgen 4096 Byte. Die ersten 2816 Zeichen bilden den Zeichensatz 1, die restlichen 1280 bilden Zeichensatz 2. Die Struktur eines Zeichensatzes wird durch deren ersten 2 Byte definiert. Das erste Byte ist die Anzahl der Byte für ein darzustellendes Zeichen, das zweite Byte die Anzahl der gültigen Bit pro Byte. Es müssen die angegebene Anzahl von Bytes für einen Zeichensatz übertragen werden, auch wenn der Zeichensatz kürzer ist.

Zeichensatz laden: ^C=<ETX>, 4096 Zeichen (Zeichensätze 3 und 4)

wie ^B, jedoch laden der Zeichensätze 3 und 4 statt 1 und 2.

3.3.4.2 Nicht im Text-Modus gültige Befehle

Punkt ändern: <ESC>, xlow, xhigh, ylow, yhigh

Ein Punkt wird durch die Koordinaten x und y adressiert. x liegt im Wertebereich (0 ... 511) und y im Bereich (0 ... 767). Zahlen in dieser Größe können binär nur in zwei Byte dargestellt werden. Die Übertragung zur VIDEO I erfolgt in der angegebenen Reihenfolge mit einem führenden <ESC> (= 1Bh oder 27d). Die Fünf-Byte-Sequenz heißt im folgenden Punktsequenz und kann mit einem führenden Umschaltzeichen auch anderen Zwecken dienen. Der Befehl setzt, löscht oder invertiert den Punkt an der angegebenen Koordinate nach Voreinstellung.

Koordinate setzen: 'K', <ESC>, xlow, xhigh, ylow, yhigh

Der Befehl setzt die Position des Grafik-Cursors. Die gespeicherte Koordinate wird von nachfolgenden Befehlen benutzt oder verändert. Die Sechs-Byte-Sequenz besteht aus dem Umschaltzeichen 'K' und einer Punktsequenz. Der Befehl ändert den Bildschirminhalt nicht.

Punkt holen: 'G', <ESC>, xlow, xhigh, ylow, yhigh

Der Befehl übergibt den Punkt an der Cursorposition (c.p.). Die Sechs-Byte-Sequenz besteht aus dem Umschaltzeichen 'G' und einer Punktsequenz. Die VIDEO I sendet darauf Offh (255d), falls der adressierte Punkt gesetzt ist. Falls er nicht gesetzt ist, sendet die VIDEO I an dieser Stelle 0h (0d).

Vektor zeichnen: 'V', <ESC>, xlow1, xhigh1, ylow1, yhigh1,
<ESC>, xlow2, xhigh2, ylow2, yhigh2

Der Befehl zeichnet eine Gerade vom angegebenen Startpunkt (x1,y1) zum angegebenen Endpunkt (x2,y2). Der Endpunkt wird neue Cursorposition. Die Voreinstellung für 'setzen', 'löschen' oder 'invertieren' wird beachtet und der Befehl dementsprechend ausgeführt. Die Elf-Byte-Sequenz besteht aus dem Umschaltzeichen 'V' und zwei Punktsequenzen.

Vektor von c.p. zeichnen: 'v', <ESC>, xlow, xhigh, ylow, yhigh

Der Befehl zeichnet eine Gerade von der Cursorposition (c.p.) zum angegebenen Punkt. Der Endpunkt wird neue Cursorposition. Die Voreinstellung für 'setzen', 'löschen' oder 'invertieren' wird beachtet und der Befehl dementsprechend ausgeführt. Die Sechs-Byte-Sequenz besteht aus dem Umschaltzeichen 'v' und einer Punktsequenz.

Byte ändern: 'S', byte

Der Befehl setzt das als Parameter angegebene Byte, beginnend mit dessen MSB, zeilenweise an die Cursorposition. Die x-Komponente der Cursor-Adresse wird um 8 erhöht.

Ist die Anzahl der Punkte bis zum Zeilenende kleiner als 8, so werden die restlichen Punkte beginnend mit dem Anfang der nächsten Zeile weitergezählt.

Die Befehlsausführung beginnt mit der Übergabe des Steuerzeichens 'S' und endet mit der anschließenden Übergabe des Datenbytes.

Byte holen: 'R'

Der Befehl übergibt acht Bildpunkte, beginnend mit der Cursorposition, in einem Byte. Der erste Punkt bildet das MSB des Bytes, dem die weiteren Punkte als Bit niedrigerer Wertigkeit desselben Bytes folgen. Die x-Komponente der Cursor-Adresse wird um 8 erhöht. Ist die Anzahl der Punkte bis zum Zeilenende kleiner als 8, so werden die restlichen Punkte beginnend mit dem Anfang der nächsten Zeile weitergezählt.

Die Befehlsausführung beginnt mit der Übergabe des Steuerzeichens 'R' und endet mit der anschließenden Übernahme des Datenbytes von der VIDEO I.

Hardcopy: 'H', <ESC>, xlow, xhigh, ylow, yhigh

Hardcopy übergibt 768 Byte aus 8 Zeilen. Die Startkoordinate muß in der ersten Spalte des Bildschirmbereiches liegen (also auf jeden Fall x = 0!). Acht untereinanderliegende Punkte werden zu einem Byte zusammengefaßt. Das Bit mit der niedrigsten Wertigkeit entspricht dem Bildpunkt aus der untersten der acht Zeilen. Die Y-Koordinate adressiert den obersten der 8 Bildpunkte. Alle 768 Byte müssen abgeholt werden, bevor eine weitere Funktion der Videokarte ausgelöst wird.

Bitte beachten Sie, daß Sie im Grafik-Modus 2 die Y-Koordinate um 16 erhöhen müssen, wenn Sie die nächsten 8 Zeilen adressieren wollen, da in dieser Betriebsart alle übertragenen Y-Koordinaten durch 2 dividiert werden!

Bei dem direkten Transfer eines Bytes von der Videokarte zum Drucker über einen BDOS-Aufruf kann es zu einem Datenverlust kommen, falls die Videokarte als Konsole eingesetzt ist! Bitte beachten Sie das entsprechende Kapitel über die 'VIDEO I als Konsole unter CP/M'!

Text-Modus einschalten: <CR>, 'C', 'c', 'D', 'd', 'F' und 'f'

Der Befehl schaltet in den Text-Modus. Nach dem Befehl sind nur noch Text-Befehle gültig. Mit der Umschaltung in den Text-Modus wird gleichzeitig einer der 6 möglichen Zeichensätze ausgewählt:

<CR> voreingestellter Zeichensatz

'C' Zeichensatz 1
'c' Zeichensatz 2

'D' Zeichensatz 3
'd' Zeichensatz 4

'F' Zeichensatz 5 (oder 1) (Voreinstellung für Grafik-Modus 1)
'f' Zeichensatz 6 (oder 2) (Voreinstellung für Grafik-Modus 2)

Bildschirm löschen: ^J = <LF>

Der Befehl löscht den Bildschirm.

Bildschirm dunkel steuern: ^Q = <DC1>

Nach Ausführung dieses Befehls ist der Bildschirm dunkel, ohne daß der Bildinhalt gelöscht wird. Es wird lediglich die Anzeige ausgeschaltet (nur ab Rev. 2 der VIDEO I).

Bildschirm hell steuern: ^P = <DLE>

Dieser Befehl hebt die Wirkung von ^Q wieder auf, d.h., die Bildschirmanzeige wird wieder eingeschaltet.

Grafik-Modus verlassen: 'E'

Der Befehl schaltet in den Alfa-Modus um. Befehle des Grafik-Modus oder des Text-Modus sind nun nicht mehr gültig.

3.3.4.3 Nur im Text-Modus gültige Befehle

Zeichen setzen: ' '...'ß'

Eines der 96 Zeichen aus dem ASCII-Zeichensatz wird an die Cursorposition auf dem Bildschirm gesetzt.

Cursor rechts: ^I = <HT>

Der Cursor wird um ein Zeichen nach rechts versetzt. Am Zeilenende geht der Cursor an den Anfang der nächsten Zeile, am Bildende geht der Cursor an den Bildanfang. Ein Verschieben des Bildinhaltes (scroll) erfolgt nicht.

Cursor links: ^H = <BS>

Der Cursor wird um ein Zeichen nach links versetzt. Vom Zeilenanfang wandert der Cursor nach diesem Befehl an das Ende der davorliegenden Zeile. Am Bildanfang wandert der Cursor nach dem Befehl in die letzte Spalte der letzten Zeile.

Cursor hoch: ^K = <VT>

Der Cursor wird um ein Zeichen (eine Zeile) nach oben versetzt. Steht der Cursor in der obersten Zeile, so bleibt er in seiner Position.

Cursor tief: ^J = <LF>

Der Cursor wird um ein Zeichen (Zeile) nach unten versetzt. Steht der Cursor in der letzten Zeile, so steht er nach dem Befehl in der ersten Spalte der ersten Zeile.

Cursor an den Zeilenanfang: ^M = <CR>

Der Befehl setzt den Cursor an den Zeilenanfang.

Text-Modus verlassen: ^Ü = <GS> oder ^Ö = <FS>

Der Befehl schaltet vom Textmodus in den Grafik-Modus.

3.4 Beispiele

3.4.1 Setzen und Löschen von Punkten

Nachfolgend wird die Eingabe für das Setzen und Löschen mehrerer Bildpunkte beispielhaft dargestellt. Folgende Punkte sollen gesetzt werden:

P1: X1 = 12D (=000CH), Y1 = 133D (=0085H)
P2: X2 = 350D (=015EH), Y2 = 1D (=0001H)

sowie anschließend der Punkt:

P3: X3 = 33D (=0021H), Y3 = 444D (=01BCH)

gelöscht werden.

Hierzu ergibt sich folgende Bytesequenz:

```
1.   02BH          ; '+'              Voreinstellung Punkt setzen
2.   01BH          ; <ESC>
3.   00CH          ; low  X1
4.   000H          ; high X1
5.   085H          ; low  Y1
6.   000H          ; high Y1
7.   01BH          ; <ESC>
8.   05EH          ; low  X2
9.   001H          ; high X2
10.  001H          ; low  Y2
11.  000H          ; high Y2
12.  02DH          ; '-'              Voreinstellung Punkt löschen
13.  01BH          ; <ESC>
14.  021H          ; low  X3
15.  000H          ; high X3
16.  0BCH          ; low  Y3
17.  001H          ; high Y3
```

Es ist zu beachten, daß alle weiteren Eingaben von X-Y-Koordinaten gemäß Voreinstellung ein Löschen der entsprechenden Punkte bewirken.

3.4.2 Zeichnen von Vektoren und Schreiben eines Textes

Als Beispiel soll ein Dreieck mit folgenden Eck-Koordinaten gezeichnet werden:

(x1,y1) = (1,1) ; (x2,y2) = (767,200) ; (x3,y3) = (368,511)

Die gleichen Koordinaten lauten in Hex-Schreibweise:

(x1,y1) = (0001,0001) ; (x2,y2) = (02FF,00C8) ; (x3,y3) = (0170,01FF)

Zum Zeichnen des Dreieckes sind nun folgende 3 Vektorsequenzen zu senden:

```
1.   056H          ; 'V',    Einleiten einer Vektorsequenz
2.   01BH          ; <ESC>
3.   001H          ; low  X1
4.   000H          ; high X1
5.   001H          ; low  Y1
6.   000H          ; high Y1
7.   01BH          ; <ESC>
8.   0FFH          ; low  X2
9.   002H          ; high X2
10.  0C8H          ; low  Y2
11.  000H          ; high Y2

12.  056H          ; 'V',    2. Vektorsequenz
13.  01BH          ; <ESC>
14.  0FFH          ; low  X2
15.  002H          ; high X2
16.  0C8H          ; low  Y2
17.  000H          ; high Y2
18.  01BH          ; <ESC>
19.  070H          ; low  X3
20.  001H          ; high X3
21.  0FFH          ; low  Y3
22.  001H          ; high Y3

23.  056H          ; 'V',    3. Vektorsequenz
24.  01BH          ; <ESC>
25.  070H          ; low  X3
26.  001H          ; high X3
27.  0FFH          ; low  Y3
28.  001H          ; high Y3
29.  01BH          ; <ESC>
30.  001H          ; low  X1
31.  000H          ; high X1
32.  001H          ; low  Y1
33.  000H          ; high Y1
```

Wie sich das ganze kürzer fassen läßt sei an der folgenden Befehlskette gezeigt: das Dreieck wird mit Hilfe des zweiten Vektorbefehls (Linie von c.p. aus zeichnen) wieder gelöscht.

Jeder Vektorbefehl setzt die Cursorposition neu.

```
34.  02DH          ; '-'       Voreinstellung für löschen

35.  076H          ; 'v'       Vektor ab Cursorposition (1,1) s.o.
36.  01BH          ; <ESC>     nach (2ffh, 0c8h)
37.  0FFH          ; low  x2
38.  002H          ; high x2
39.  0C8H          ; low  y2
40.  000H          ; high y2

41.  076H          ; 'v'       Vektor von c.p. nach (170h, 1ffh)
42.  01BH          ; <ESC>
43.  070H          ; low  x3
44.  001H          ; high x3
45.  0FFH          ; low  y3
46.  001H          ; high y3

47.  076H          ; 'v'       Vektor von c.p. nach (1,1)
48.  01BH          ; <ESC>
49.  001H          ; low  x1
50.  000H          ; high x1
51.  001H          ; low  y1
52.  000H          ; high y1
```

Die folgende Befehlskette schreibt den Text 'Dreieck' an den Punkt (x1, y1). Sie setzt voraus, daß zuvor ein Zeichensatz geladen wurde.

```
53.  063H          ; 'c'  Text-Modus (Zeichensatz 2) einschalten
54.  044H          ; 'D'
55.  072H          ; 'r'
56.  065H          ; 'e'
57.  069H          ; 'i'
58.  065H          ; 'e'
59.  063H          ; 'c'
60.  06BH          ; 'k'
61.  01DH          ; ^Ü   Text-Modus wieder ausschalten
```

4. Tastatur-Funktionen

4.1 Auslösen von VIDEO I-Funktionen über die Tastatur

Viele der internen Funktionen von VIDEO I lassen sich sowohl vom Computer als auch direkt von der Tastatur auslösen. Zum Auslösen von 'Tastatur-Funktionen' werden nur Tastencodes mit gesetztem 8. Bit benutzt. Falls die verwendete Tastatur nicht über Funktionstasten mit gesetztem 8. Bit verfügt, muß zuerst vom Computer über die Funktion 56 eine Tastaturumcodierung vorgenommen werden.

Die nachfolgenden Funktionen können auch über die Tastatur direkt ausgelöst werden. Dabei ist zu beachten, daß bei Angabe mehrerer Tasten, diese alle hintereinander betätigt werden müssen, um die gewünschte Funktion auszulösen.

Die Tastenbezeichnung gilt für die meisten Tastaturen mit zusätzlichen Funktionstasten (z.B. Cherry, Marquard). Im Zweifelsfall ist der Tastencode zu vergleichen (siehe Kurzübersicht im Anhang). Bei einem abweichenden Tastencode kann eine Codegleichheit über die Funktion 56 vom Computer aus realisiert werden.

'Blättern' in zurückliegenden Bildschirmseiten:

Funktion 29:	**ZEILE ZURÜCK**	Taste: F7
Funktion 30:	**ZEILE VOR**	Taste: F6
Funktion 31:	**AKTUELLE BILDSCHIRMSEITE EINSTELLEN**	Taste: F4
Funktion 32:	**SEITE ZURÜCK**	Taste: F3
Funktion 33:	**SEITE VOR**	Taste: F2

Die Funktionen 29 bis 33 verhalten sich etwas anders, wenn sie direkt von der Tastatur ausgelöst werden:

Bei Funktion 29 und 30 gehen keine Zeilen verloren; jede mit Funktion 29 herausgeschobene Zeile wird mit Funktion 30 wieder zurückgeholt. Funktion 30 kann jedoch nur dann ausgelöst werden, wenn die aktuelle Bildschirmseite nicht aktiv ist. Die aktuelle Bildschirmseite ist inaktiv, wenn um mindestens eine Zeile zurückgegangen wurde.

Nur wenn die aktuelle Bildschirmseite eingestellt ist, nimmt VIDEO I vom Computer Zeichen an, d.h., beim 'Zurückblättern' auf vorherige Bildschirmseiten wird die Ausgabe auf die VIDEO-Platine unterbrochen. Durch wechselseitiges Betätigen der Funktionen 29, 30, 32 und 33 kann die aktuelle Bildschirmseite wieder erreicht werden. Mit Funktion 31 wird die aktuelle Bildschirmseite sofort wieder eingestellt. Jede andere Eingabe von der Tastatur sendet das Zeichen zum Computer und hat zur Nebenwirkung, daß ebenfalls sofort die aktuelle Bildschirmseite wieder eingestellt wird.

Die Funktionen 29 bis 33 können beliebig miteinander kombiniert werden. Wenn mit Funktion 32 keine weitere Seite mehr zurückgeblättert werden kann, ist es meist noch möglich, mit Funktion 29 einige Zeilen zurückzugehen.

Kaltstart auslösen:

Funktion 41: **RESET VIDEO I** Tasten: **F1,F1,F13**

Terminal-Emulation wählen:

Funktion 46: **TERMINAL 4 AKTIVIEREN** Tasten: **F1,F5**
Funktion 47: **TERMINAL 3 AKTIVIEREN** Tasten: **F1,F4**
Funktion 48: **TERMINAL 2 AKTIVIEREN** Tasten: **F1,F3**
Funktion 49: **TERMINAL 1 AKTIVIEREN** Tasten: **F1,F2**

Bildschirmformat einstellen:

Funktion 50: **24 ZEILEN MIT 132 ZEICHEN PRO ZEILE** Tasten: **F1,F1,F4**
Funktion 51: **24 ZEILEN MIT 96 ZEICHEN PRO ZEILE** Tasten: **F1,F1,F3**
Funktion 52: **24 ZEILEN MIT 80 ZEICHEN PRO ZEILE** Tasten: **F1,F1,F2**
Funktion 100: **25 ZEILEN MIT 132 ZEICHEN PRO ZEILE** Tasten: **F1,F1,F12**
Funktion 101: **25 ZEILEN MIT 96 ZEICHEN PRO ZEILE** Tasten: **F1,F1,F11**
Funktion 102: **25 ZEILEN MIT 80 ZEICHEN PRO ZEILE** Tasten: **F1,F1,F10**

Zwischen deutschem und amerikanischem Zeichensatz wählen:

Funktion 53: **US - ZEICHENSATZ** Tasten: **F1,F1,F6**
Funktion 54: **DEUTSCHER ZEICHENSATZ** Tasten: **F1,F6**

Inversen Bildschirm ein- und ausschalten:

Funktion 81: **INVERSER BILDSCHIRM AUS** Tasten: **F1,F8**
Funktion 82: **INVERSER BILDSCHIRM EIN** Tasten: **F1,F1,F8**

Tastatur wieder einschalten:

Funktion 84: **TASTATUR EINSCHALTEN** (keyboard unlock)

Falls vom Computer die Tastatur ausgeschaltet wurde, so kann sie mit der gleichen Codefolge sowohl vom Computer als auch von der Tastatur wieder eingeschaltet werden. Die gültige Codefolge hängt von der jeweiligen Terminal-Emulation ab.

Blinkenden oder nicht blinkenden Cursor einstellen:

Funktion 86: **CURSOR BLINKEN EIN** Tasten: **F1,F9**
Funktion 87: **CURSOR EIN** Tasten: **F1,F1,F9**

Tastaturpuffer einstellen:

Funktion 89: **TASTATUR-PUFFER DEAKTIVIEREN** Tasten: **F1,F1,F7**
Funktion 90: **TASTATUR-PUFFER AKTIVIEREN** Tasten: **F1,F7**
Funktion 91: **TASTATUR-PUFFER LÖSCHEN** Taste: **F5**

Testbild anzeigen:

Funktion 92: **TEST-BILD AUSGEBEN** Tasten: **F1,F1,F5**

4.2 Änderung der Tastatur-Funktions-Steuertabelle

Die VIDEO I-Firmware enthält drei Steuertabellen für die Tastencode-Belegung der einzelnen Steuerfunktionen. Diese drei Tabellen sind im Anhang vollständig abgedruckt.

Die in Kapitel 4.1 beschriebenen Funktionen gelten für die Belegung der drei Steuertabellen wie im Anhang abgedruckt. Für bestimmte Tastaturen (oder aus anderen Gründen) kann es jedoch wünschenswert sein, die einzelnen Funktionen über andere Zeichenfolgen auszulösen. In einigen Fällen ist es auch denkbar, daß auf bestimmte Tastaturfunktionen vollständig verzichtet werden soll, z.B. um Fehlbedienung durch ungeübtes Personal zu verhindern.

Mit Hilfe der Funktion 56 kann die Funktionstastenzuordnung vom Computer aus temporär verändert werden. Zum dauerhaften Verändern der Funktionsauslösung ist eine Änderung der Steuertabellen in der Firmware der VIDEO I notwendig.

Tabelle 1 enthält alle Funktionen, die mit nur einem Tastendruck ausgelöst werden können. Außerdem ist in Tabelle 1 auch der Tastencode enthalten, mit dem auf die Tabelle 2 weitergeschaltet wird.

Tabelle 2 enthält alle Funktionen, die mit 2 Tastenbetätigungen ausgelöst werden können. Der 1. Tastendruck ist dabei für alle Funktionen der Tabelle 2 gleich. Außerdem enthält Tabelle 2 den Tastencode, mit dem auf die Tabelle 3 weitergeschaltet wird.

Tabelle 3 enthält alle Funktionen, die mit 3 Tastenbetätigungen ausgelöst werden können. Die ersten beiden Tastendrücke sind dabei für alle Funktionen der Tabelle 3 gleich.

In allen 3 Tabellen bedeutet ein Code mit gesetztem 8. Bit, daß die gewünschte Funktion bei Betätigung dieser Taste ausgelöst wird. Ein Code, bei dem das 8. Bit nicht gesetzt ist, hat zur Folge, daß diese Funktion mit dieser Tastenfolge nicht ausgelöst werden kann.

Beispiel: Für die Funktion 'CURSOR ON' ist in Tabelle 1 und 2 eine Null und in Tabelle 3 eine 88H eingetragen. Diese Funktion kann deshalb nur als 3-Byte-Sequenz von der Tastatur ausgelöst werden. Falls diese Funktion auch als 1-Byte-Sequenz mit dem Code 93H ausgelöst werden soll, muß in Tabelle 1 auf Adresse 19AAH der Code 93H eingetragen werden. Falls die Funktionsauslösung mit der bisherigen 3-Byte-Sequenz nicht mehr möglich sein soll, muß in Tabelle 3 auf Adresse 19EAH eine Null eingetragen werden.

5. Terminal-Emulationen

Im Maschinenprogramm von VIDEO I erfolgt der Aufruf aller Steuerfunktionen unter Benutzung einer Codewort-Tabelle. Sie gliedert sich in drei Untertabellen zur Unterscheidung von 1-, 2- und 3-Byte-Befehlen. Es sind 4 solcher Codewort-Tabellen vorhanden, von denen jede für sich eine bestimmte Terminal-Emulation repräsentiert. Eine 5. Codewort-Tabelle kann vom Computer in die VIDEO I geladen werden.

5.1 Aktivieren einer implementierten Terminal-Emulation

Zur Aktivierung einer Terminal-Emulation ist eine der nachfolgenden Funktionen auszulösen:

Funktion 45: **TERMINAL 5 AKTIVIEREN**
Funktion 46: **TERMINAL 4 AKTIVIEREN**
Funktion 47: **TERMINAL 3 AKTIVIEREN**
Funktion 48: **TERMINAL 2 AKTIVIEREN**
Funktion 49: **TERMINAL 1 AKTIVIEREN**

Die Tabelle für Terminal 1 ist mit der Emulation des ADDS VIEWPOINT belegt worden.

Die Tabelle für Terminal 2 ist mit der Emulation eines GEVEKE VISA 30/40 belegt worden.

Die Tabelle für Terminal 3 ist mit der Emulation eines erweiterten ADM 3a (ähnlich info-S Video 7) belegt worden.

Die Tabelle für Terminal 4 ist nur mit OFFH vorbesetzt und kann vom Anwender im EPROM für eine gewünschte Terminal-Emulation nachprogrammiert werden.

Die Tabelle für Terminal 5 muß vorher vom Computer aus geladen werden. Bitte beachten Sie dazu Kapitel 5.2.

5.2 Implementierung einer neuen Terminal-Emulation

Eine neue Terminal-Emulation kann durch Ändern der Emulationstabellen im EPROM (z.B. der Tabellen für Terminal 4) oder durch Laden der Emulationstabellen vom Computer aus realisiert werden. Zum Laden der Emulationstabellen vom Computer aus steht folgende Funktion zur Verfügung:

Funktion 44: **TERMINALEMULATION FÜR TERMINAL 5 LADEN**

Nach Auslösung dieser Funktion muß eine vollständige Emulationstabelle in die VIDEO I geladen werden. Eine vollständige Emulationstabelle besteht aus 3 Codetabellen mit einer Länge von jeweils 128 Bytes. Die Anzahl der einzugebenen Bytes beträgt also insgesamt 384 (3 x 128). Der Aufbau einer Emulationstabelle kann dem Anhang

entnommen werden. Nach Übertragung des letzten Bytes der Emulationstabelle ist diese Funktion automatisch beendet.

Dabei ist jedoch zu beachten:

- Eine Funktion ist nicht belegt bei Eintrag des Wertes 00H.
- In Tabelle 1 (Control table) sind nur Control-Zeichen zugelassen, deren Wertigkeit also kleiner 20H=32D ist.
- In Tabelle 2 (byte sequence table) und 3 (two byte sequence table) sind alle Wertigkeiten von 00H bis 0FFH zugelassen.

Mit Aufruf von Funktion 45 wird die Codetabelle eingeschaltet. Der Zugriff auf diese Tabelle ist jedoch langsamer, da er nur in den Austastlücken möglich ist.

Beispiel für ein Programm zum Laden und Aktivieren der Emulation für Terminal 5:

```
        TITLE   'LADEN UND AKTIVIEREN VON TERMINAL 5 -- TERM5 1.001'
        MACLIB Z80

;*****************************************************************
;*                                                               *
;*      LADEN UND AKTIVIEREN DES TERMINALS 5                     *
;*                                                               *
;*****************************************************************


TERM5:  LXI     H,START         ; TABELLEN-START
        LXI     B,LENGTH        ; TABELLEN-LAENGE
LOOP:   CALL    VIDOUT          ; CHAR ZUR VIDEO-KARTE
        INX     H               ; NEXT CHAR
        DCX     B               ; COUNT DOWN
        MOV     A,B
        ORA     C               ; 0 ?
        JRNZ    LOOP            ; SCHLEIFE, BIS ALLE ZEICHEN AUSGEGEBEN SIND
        RET

START:  DB      ESC
        DB      'X'
        DB      44              ; LOAD TERMINAL 5

;-------------------------------------------------------
;       TERMINAL 5: TABELLE 1, 2 und 3
;-------------------------------------------------------

TABLE5:
```

hier folgen nun die 3 Tabellen zu je 128Bytes (siehe Anhang E)

```
        DB      ESC
        DB      'X'
```

```
        DB      45              ; ENABLE TERMINAL 5

LENGTH  EQU     $-START         ; TABELLENLAENGE

;*****************************************************************
;*                                                               *
;*      EIN/AUSGABE VON ZEICHEN ZUR VIDEO1      1.001            *
;*                                                               *
;*      ERSTELLT AM:    24.09.84        DURCH: D. JANICH         *
;*                                                               *
;*****************************************************************

;*****************************************************************
;*                                                               *
;*      PARAMETER:                                               *
;*      IN:     HL = .CHAR                                       *
;*                                                               *
;*      ZERSTOERTE REGISTERINHALTE: AF                           *
;*                                                               *
;*****************************************************************

VIDCNT  EQU     0B9H    ; VIDEO CONTROL
VIDDAT  EQU     0B8H    ; VIDEO DATA

VIDOUT:

VIDPOL: IN      VIDCNT
        ANI     02              ; STATUS BIT MASKIEREN
        JRNZ    VIDPOL          ; SCHLEIFE, BIS VIDEO1 READY

        MOV     A,M             ; OUTPUT CHAR
        OUT     VIDDAT          ; ZUR VIDEO-KARTE

        RET

        END
```

Das obige Programm kann nach Einfügen der Emulationstabelle mit dem RMAC-Assembler (gehört zum CP/M PLUS) assembliert werden.

6. Steuerfunktionen in numerischer Reihenfolge

Funktion 1: **ZWEI-BYTE-SEQUENZ** (2 byte sequence)

Dies ist eine Vorbereitungsfunktion einer 2-Byte-Sequenz. Üblicherweise benutzt man hier das ESCAPE (<ESC>=1BH=27D) Control-Zeichen. Dieses Zeichen muß in der Control-Tabelle (siehe Anhang) als letztes Byte eingetragen sein. Der auf <ESC> folgende Code bestimmt den Aufruf der Funktion und ist in der Tabelle für 3-Byte-Sequenzen eingetragen.

Funktion 2: **DREI-BYTE-SEQUENZ** (3 byte sequence)

Bei dieser Funktion handelt es sich um eine Vorbereitungsfunktion zu einer 3-Byte-Sequenz. Diese Funktion kann nur nach Aufruf von Funktion 1 (ZWEI-BYTE-SEQUENZ) erreicht werden. Beispielsweise wird diese Funktion in der Voreinstellung (Terminal 1) durch <ESC> 'X' aufgerufen. Der Eintrag 'X' muß in der 2-Byte-Sequenz-Tabelle erfolgt sein. Der auf <ESC> 'X' folgende Code bestimmt den Aufruf der Funktion und ist in der Tabelle für 3-Byte-Sequenzen eingetragen.

Funktion 3: **CURSOR ZURÜCK** (backspace)

Der Cursor rückt um eine Position zurück. Stand der Cursor am Beginn einer Zeile, so rückt er an das Ende der vorhergehenden Zeile. Bei der Ausgangsposition am Anfang der ersten Zeile wird vorher ein "scroll down" durchgeführt.

Funktion 4: **ZEILENVORSCHUB** (linefeed)

Der Cursor belegt die nächste Zeile; seine Spaltenposition bleibt erhalten. Ist die Ausgangsposition des Cursors die letzte Zeile des Bildschirms, erfolgt zuvor ein "scroll up" (Funktion 30), so daß der Cursor wieder in der letzten Zeile zu finden ist.

Funktion 5: **UMGEKEHRTER ZEILENVORSCHUB** (reverse linefeed)

Der Cursor belegt die vorliegende Zeile; seine Spaltenposition bleibt erhalten. Ist die Ausgangsposition des Cursors die erste Zeile des Bildschirms, erfolgt zuvor ein scroll down (Funktion 29). Der Cursor befindet sich in diesem Fall nach Ausführen der Funktion wieder in der ersten Zeile.

Funktion 6: **CARRIAGE RETURN**

Der Cursor rückt an den Anfang der Zeile.

Funktion 7: **CARRIAGE RETURN + LINEFEED**

Es werden nacheinander die Funktionen 4 und 6 ausgeführt.

Funktion 8: **CARRIAGE RETURN UND REST DER ZEILE LÖSCHEN**
(carriage return and erase end of line)

Es wird der Rest der Zeile ab Cursorposition (einschließlich) gelöscht, und der Cursor rückt an den Anfang der Zeile. Falls der Cursor sich bei Aufruf der Funktion bereits am Anfang einer Zeile befindet, erfolgt keine Aktion, d.h., auch der Rest der Zeile wird nicht gelöscht.

Funktion 9: **ABSOLUTE CURSOR ADRESSIERUNG MIT OFFSET (Y,X)**
(absolute cursor addressing with offset (y,x))

Der Cursor kann an eine beliebige Stelle des Bildschirms gesetzt werden. Die Adressierung erfolgt über eine Sequenz von mindestens 3 Zeichen:

I. Voreinstellung Cursor Adressierung - 1 bis 3 Byte Sequenz
II. Vertikale und horizontale Adresse - 2 Byte
Als erstes Adreßbyte wird die Y-Adresse (Zeilennummer) und als zweites Adreßbyte die X-Adresse (Spaltennummer) übertragen. Die Numerierung beginnt mit der Zeile 0 und der Spalte 0!

Es ist zu beachten, daß der X- und Y-Wert bei dieser Funktion mit einem Offset von 32 = 20H (BLANK) erwartet wird.

Zeile 0 und Spalte 1 sind somit mit der Sequenz:

<ESC> 'X' 9 ' ' '!'

adressierbar.

Funktion 10: **ABSOLUTE CURSOR ADRESSIERUNG OHNE OFFSET (Y,X)**
(absolute cursor addressing without offset (y,x))

Der Cursor kann an eine beliebige Stelle des Bildschirms gesetzt werden. Die Adressierung erfolgt über eine Sequenz von mindestens 3 Zeichen:

I. Voreinstellung Cursor Adressierung - 1 bis 3 Byte Sequenz
II. Vertikale und horizontale Adresse - 2 Byte
Als erstes Adreßbyte wird die Y-Adresse (Zeilennummer) und als zweites Adreßbyte die X-Adresse (Spaltennummer) übertragen. Die Numerierung beginnt mit der Zeile 0 und der Spalte 0!

Es ist zu beachten, daß der X- und Y-Wert bei dieser Funktion ohne Offset erwartet wird.

Zeile 0 und Spalte 1 sind somit mit der Sequenz:

<ESC> 'X' 10 0 1

adressierbar.

Funktion 11: **ABSOLUTE CURSOR ADRESSIERUNG MIT OFFSET (X,Y)**
(absolute cursor addressing with offset (x,y))

Der Cursor kann an eine beliebige Stelle des Bildschirms gesetzt werden. Die Adressierung erfolgt über eine Sequenz von mindestens 3 Zeichen:

I. Voreinstellung Cursor Adressierung - 1 bis 3 Byte Sequenz
II. Horizontale und vertikale Adresse - 2 Byte
Als erstes Adreßbyte wird die X-Adresse (Spaltennummer) und als zweites Adreßbyte die Y-Adresse (Zeilennummer) übertragen. Die Numerierung beginnt mit der Zeile 0 und der Spalte 0!

Es ist zu beachten, daß der X- und Y-Wert bei dieser Funktion mit einem Offset von 32 = 20H (BLANK) erwartet wird.

Zeile 0 und Spalte 1 sind somit mit der Sequenz:

<ESC> 'X' 11 '!' ' '

adressierbar.

Funktion 12: **ABSOLUTE CURSOR ADRESSIERUNG OHNE OFFSET (X,Y)**
(absolute cursor addressing without offset (x,y))

Der Cursor kann an eine beliebige Stelle des Bildschirms gesetzt werden. Die Adressierung erfolgt über eine Sequenz von mindestens 3 Zeichen:

I. Voreinstellung Cursor Adressierung - 1 bis 3 Byte Sequenz
II. Horizontale und vertikale Adresse - 2 Byte
Als erstes Adreßbyte wird die X-Adresse (Spaltennummer) und als zweites Adreßbyte die Y-Adresse (Zeilennummer) übertragen. Die Numerierung beginnt mit der Zeile 0 und der Spalte 0!

Es ist zu beachten, daß der X- und Y-Wert bei dieser Funktion ohne Offset erwartet wird.

Zeile 0 und Spalte 1 sind somit mit der Sequenz:

<ESC> 'X' 12 1 0

adressierbar.

Funktion 13: **HORIZONTALE CURSOR ADRESSIERUNG**
(horizontal cursor addressing)

Der Cursor kann an eine beliebige Stelle innerhalb seiner Zeile positioniert werden. Die Adressierung erfolgt über eine Sequenz von mindestens 2 Zeichen:

I. Voreinstellung horizontale Cursor-Adressierung - 1 bis 3 Byte Sequenz
II. Horizontale Adresse (Spaltennummer) - 1 Byte

Es ist zu beachten, daß die horizontale Adresse bei dieser Funktion mit einem Offset von 32 = 20H (BLANK) erwartet wird. Die Spaltennumerierung beginnt mit der Nummer 0.

Spalte 0 ist somit mit der Sequenz:

<ESC> 'X' 13 ' '

adressierbar.

Funktion 14: **VERTIKALE CURSOR ADRESSIERUNG**
(vertical cursor addressing)

Der Cursor kann an eine beliebige Stelle innerhalb seiner Spalte positioniert werden. Die Adressierung erfolgt über eine Sequenz von mindestens 2 Zeichen:

I. Voreinstellung vertikale Cursor Adressierung - 1 bis 3 Byte Sequenz
II. Vertikale Adresse (Zeilennummer) - 1 Byte

Es ist zu beachten, daß die vertikale Adresse bei dieser Funktion mit einem Offset von 32 = 20H (BLANK) erwartet wird. Die Zeilennumerierung beginnt mit der Nummer 0.

Zeile 0 ist somit mit der Sequenz:

<ESC> 'X' 14 ' '

adressierbar.

Funktion 15: **RESERVIERT FÜR ZUKÜNFTIGE ERWEITERUNGEN**
(reserved for future use)

Funktion 16: **RESERVIERT FÜR ZUKÜNFTIGE ERWEITERUNGEN**
(reserved for future use)

Funktion 17: **CURSOR NACH UNTEN** (cursor down)

Der Cursor rückt um eine Zeile nach unten. Stand der Cursor in der letzten Zeile, bleibt seine Position unverändert.

Funktion 18: **CURSOR NACH OBEN** (cursor up)

Der Cursor rückt um eine Zeile nach oben. Stand der Cursor in der ersten Zeile, bleibt seine Position unverändert.

Funktion 19: **CURSOR LINKS** (cursor left)

Der Cursor rückt um eine Position nach links. Stand der Cursor am Beginn einer Zeile, bleibt seine Position unverändert.

Funktion 20: **CURSOR RECHTS** (cursor right)

Der Cursor rückt um eine Position nach rechts. Stand der Cursor am Ende einer Zeile, bleibt seine Position unverändert.

Funktion 21: **CURSOR VOR** (cursor forward)

Der Cursor rückt um eine Position weiter. Stand der Cursor am Ende einer Zeile, so rückt er an den Beginn der nächsten Zeile. Bei der Ausgangsposition am Ende der letzten Zeile wird vorher ein "scroll up" durchgeführt.

Funktion 22: **ZEICHEN AN CURSOR-POSITION SENDEN**
(send char at cursor)

VIDEO I sendet das augenblicklich unter dem Cursor stehende Zeichen an den Computer zurück. Falls dieses Zeichen nicht dem augenblicklichen Zeichensatz angehört, sendet VIDEO I eine 0 oder einen Code zwischen 0 und BLANK (ausschließlich) zurück. Dies ist zum Beispiel dann der Fall, wenn der deutsche Zeichensatz aktiv ist, der Cursor aber auf einer eckigen Klammer steht, die unter dem amerikanischen Zeichensatz geschrieben wurde. Bei gesetztem Underline-Attribut sendet VIDEO I das augenblickliche Zeichen mit gesetztem 8. Bit zurück.

Bei allen Bildschirmformaten mit 40, 48 oder 66 Zeichen pro Zeile und/oder 12 Zeilen pro Bildschirmseite sendet VIDEO I bei dieser Funktion immer eine 0 zurück!

Funktion 23: **CURSOR-ADRESSE MIT OFFSET SENDEN (Y,X)**
(send cursor address with offset (y,x))

VIDEO I sendet die augenblickliche Cursor-Adresse an den Computer zurück. Dabei wird zuerst die Y-Adresse (Zeilennummer = 1 Byte) und anschließend die X-Adresse (Spaltennummer = 1 Byte) übertragen. Die Numerierung beginnt auch hier mit Zeile 0 und Spalte 0. Auf die Y- bzw. X-Adresse addiert VIDEO I einen Offset von 20H (=BLANK).

Funktion 24: **CURSOR-ADRESSE OHNE OFFSET SENDEN (Y,X)**
(send cursor address without offset (y,x))

VIDEO I sendet die augenblickliche Cursor-Adresse an den Computer zurück. Dabei wird zuerst die Y-Adresse (Zeilennummer = 1 Byte) und anschließend die X-Adresse (Spaltennummer = 1 Byte) übertragen. Die Numerierung beginnt auch hier mit Zeile 0 und Spalte 0. Auf die Y- bzw. X-Adresse addiert VIDEO I keinen Offset.

Funktion 25: **CURSOR-ADRESSE MIT OFFSET SENDEN (X,Y)**
(send cursor address with offset (x,y))

VIDEO I sendet die augenblickliche Cursor-Adresse an den Computer zurück. Dabei wird zuerst die X-Adresse (Spaltennummer = 1 Byte) und anschließend die Y-Adresse (Zeilennummer = 1 Byte) übertragen. Die Numerierung beginnt auch hier mit Zeile 0 und Spalte 0. Auf die Y- bzw. X-Adresse addiert VIDEO I einen Offset von 20H (=BLANK).

Funktion 26: **CURSOR-ADRESSE OHNE OFFSET SENDEN (X,Y)**
(send cursor address without offset (x,y))

VIDEO I sendet die augenblickliche Cursor-Adresse an den Computer zurück. Dabei wird zuerst die X-Adresse (Spaltennummer = 1 Byte) und anschließend die Y-Adresse (Zeilennummer = 1 Byte) übertragen. Die Numerierung beginnt auch hier mit Zeile 0 und Spalte 0. Auf die Y- bzw. X-Adresse addiert VIDEO I keinen Offset.

Funktion 27: **TEILSCROLL ABWAERTS** (partial scroll down)
bzw. **ZEILE EINFUEGEN** (insert line)

Der mit Beginn der Cursorzeile dargestellte untere Bildinhalt wird um eine Zeile nach unten geschoben sowie die Cursorzeile gelöscht. In dem nach unten verschobenen Bildschirmteil werden bei der Revision 1.0 der VIDEO I-Platine die Attribute "blinken", "invers" und "halbe Intensität" gelöscht. Ab Revision 2.0 der Platine werden diese Attribute mit verschoben. Der obere Teil des Bildschirms bleibt unverändert. Da bei dieser Funktion eine leere Zeile innerhalb des Textes entsteht, wird sie auch ZEILE EINFUEGEN (insert line) genannt.

Funktion 28: **TEILSCROLL AUFWAERTS** (partial scroll up)

Die der Cursorzeile folgende untere Bildschirmhälfte wird um eine Zeile nach oben geschoben sowie am unteren Bildrand eine Leerzeile eingefügt. Der Inhalt der Cursorzeile wird dabei überschrieben und geht verloren. Die Attribute "blinken", "invers" und "halbe Intensität" werden bei der Revision 1.0 der VIDEO I-Platine in der unteren Bildschirmhälfte gelöscht. Ab Revision 2.0 der Platine werden diese Attribute mit verschoben. Der obere Teil des Bildschirms bleibt unverändert.

Funktion 29: **SCROLL NACH UNTEN** (scroll down)

Der gesamte sichtbare Text wird um eine Zeile nach unten geschoben. Hierbei geht der Inhalt der zuvor sichtbaren letzten Zeile verloren. Der Cursor wandert mit dem Text nach unten, bleibt also bezogen auf den dargestellten Text an der gleichen Position. War der Cursor vor Ausführung des "scroll down" in der letzten Zeile des Bildschirms, ist er auch nach Ausführung des "scroll down" in der letzten Zeile zu finden. In der obersten Zeile erscheint beim scroll nach unten die letzte Zeile der vorherigen Bildschirmseite, d.h. die zuletzt beim scroll nach oben herausgeschobene Zeile.

Anmerkung: Die bei dieser Funktion nach unten herausgeschobene Zeile erscheint verstümmelt auf der am weitesten zurückliegenden Bildschirmseite, wenn auf diese zurückgeblättert wird.

Falls die Funktion 29 vom Computer 'zu oft' hintereinander aufgerufen wird (abhängig vom Bildschirmformat) erscheinen nach einiger Zeit die nach unten herausgeschobenen Zeilen verstümmelt wieder im Bild. Bei einem Bildschirmformat von 24x80 ist dies nach ca. 180 Aufrufen der Funktion 29 ohne zwischenzeitlichen 'scroll up' der Fall. Die Anzahl der erlaubten 'scroll down' hintereinander ergibt sich aus der Anzahl der gespeicherten vergangenen Bildschirmseiten und der Anzahl der Zeilen pro Bildschirmseite.

Die obige Anmerkung gilt nur, falls die Funktion 29 vom Computer ausgelöst wird. Bei Auslösung direkt von der Tastatur tritt der obige Effekt nicht auf.

Funktion 30: **SCROLL NACH OBEN** (scroll up)

Der gesamte sichtbare Text wird um eine Zeile nach oben geschoben. Am unteren Bildrand rückt gleichzeitig eine Leerzeile nach. War der Cursor vor Ausführung des "scroll up" in der ersten Zeile des Textes, ist er auch nach erfolgtem scroll in der ersten Zeile zu finden, anderenfalls rückt er mit dem Text nach oben. Bezogen auf den dargestellten Text bleibt der Cursor dann an der gleichen Stelle.

Funktion 31: **AKTUELLE BILDSCHIRMSEITE EINSTELLEN** (page actual)

Falls mit den Funktionen 32 und 33 'geblättert' wurde, kann mit dieser Funktion die vor dem Blättern aktive (=aktuelle) Bildschirmseite wieder eingestellt werden. Falls die aktuelle Bildschirmseite bereits eingestellt ist, bleibt der Aufruf dieser Funktion ohne Wirkung.

Funktion 32: **SEITE ZURÜCK** (page down)

Der mit "scroll up"-Funktionen durch den Bildschirm geschobene Text kann durch diese Funktion wieder sichtbar gemacht werden. Es werden jeweils die 12/24 oder 25 (je nach Anzahl der Zeilen pro Bildschirmseite) vorhergehenden Zeilen (= eine Seite) sichtbar. Dies kann bis zu 7mal erfolgen, da maximal 7 vorangegangene Seiten gespeichert werden (24x80-Mode). (Im 24x96-Mode werden nur 6 und im 24x132-Mode nur 4 vorangegangene Seiten gespeichert.) Die Bildschirmposition des Cursors bleibt unverändert. Falls weiter als die gespeicherten Seiten zurückgeblättert wurde, erscheint wieder die aktuelle Bildschirmseite.

Anmerkung: Es ist möglich, in eine zurückliegende Seite neuen Text einzuschreiben. Solange in dieser Seite kein "scroll up" stattfindet, ist es auch möglich, wieder in die aktuelle Seite zurückzukehren. Sobald jedoch ein "scroll up" in dieser zurückliegenden Seite stattgefunden hat, wird diese Seite zur aktuellen Bildschirmseite!

Funktion 33: **SEITE VOR** (page up)

Falls mit Funktion 32 zurückgeblättert wurde, kann mit dieser Funktion wieder nach vorne geblättert werden. Falls die aktuelle Bildschirmseite aktiv ist, bleibt diese Funktion ohne Wirkung.

Funktion 34: **ZEILE LÖSCHEN** (erase line)

Die Zeile des Cursors wird gelöscht. Dabei findet jedoch kein Scrolling statt. Nach Ausführung dieser Funktion befindet sich also eine leere Zeile auf dem Bildschirm.

Funktion 35: **ENDE DER ZEILE LÖSCHEN** (erase end of line)

Das Ende der Cursorzeile wird gelöscht. Die Cursorposition ist hierbei eingeschlossen.

Funktion 36: **CURSOR HOME**

Der Cursor rückt an den Beginn der ersten Zeile.

Funktion 37: **BILDSCHIRM LÖSCHEN UND CURSOR HOME**
(erase screen and cursor home)

Der gesamte Bildschirm wird gelöscht und der Cursor an den Beginn der ersten Zeile gesetzt.

Funktion 38: **BILDSCHIRM LÖSCHEN** (erase screen)

Der Bildschirm wird gelöscht. Die Position des Cursors bleibt unverändert.

Funktion 39: **ANFANG DES BILDSCHIRMS LÖSCHEN** (erase start of screen)

Der Anfang des Bildschirms bis zur Cursorposition ausschließlich wird gelöscht.

Funktion 40: **ENDE DES BILDSCHIRMS LÖSCHEN** (erase end of screen)

Das Ende des Bildschirms ab Cursorposition einschließlich wird gelöscht.

Funktion 41: **RESET VIDEO I**

Nach Aufruf dieser Funktion wird ein Kaltstart durchgeführt und alle internen Parameter auf die nach dem Einschalten der VIDEO I gültigen Werte zurückgesetzt:

- 24 Zeilen mit 80 Zeichen pro Zeile
- Bildschirm nicht invertiert
- Cursor blinkend
- alle Strings gelöscht
- keine Zeichenumsetzung für Bildschirm bzw. Tastatur
- amerikanischer Zeichensatz
- alle Attribute gelöscht
- Emulation für Terminal 1 aktiv
- Sign-On-Message in Zeile 0
- Cursor in Zeile 1 und Spalte 0
- Tastaturpuffer aktiv (aber leer)
- Bildschirmanzeige eingeschaltet
- Cursor-Stack leer

Funktion 42: **GRAFIK MODE 2** (grafic mode 2)

VIDEO I wird in den Grafik-Modus mit 768 x 256 Bildpunkten geschaltet (siehe Kapitel 3). Falls die Zeichendarstellung auf dem Bildschirm ausgeschaltet war (Funktion 99), so wird sie gleichzeitig wieder eingeschaltet.

Funktion 43: **GRAFIK MODE 1** (grafic mode 1)

VIDEO I wird in den Grafik-Modus mit 768 x 512 Bildpunkten geschaltet (siehe Kapitel 3). Falls die Zeichendarstellung auf dem Bildschirm ausgeschaltet war (Funktion 99), so wird sie gleichzeitig wieder eingeschaltet.

Funktion 44: **TERMINALEMULATION FÜR TERMINAL 5 LADEN**
(load terminal 5)

Die im EPROM gespeicherten Terminal-Emulationen (Terminal 1-4) können durch eine Emulation im RAM erweitert werden. Hierzu müssen alle 3 Codetabellen geladen werden. Die Anzahl der einzugebenen Bytes beträgt insgesamt 384 (3 x 128). Startposition ist der Beginn der Control-Code-Tabelle (siehe Kapitel 5.2).

Funktion 45: **TERMINAL 5 AKTIVIEREN**

Zur Interpretation der Terminal-Funktionen wird nach Ausführung dieser Funktion die Terminal-Tabelle Nr. 5 benutzt. Diese Tabelle muß vorher vom Anwender geladen worden sein (siehe Funktion 44). Wegen des RAM-Zugriffs nur während der Austastlücken des BAS-Signales ist die Verwendung dieser Terminal-Emulation langsamer als die der übrigen Tabellen im EPROM.

Funktion 46: **TERMINAL 4 AKTIVIEREN**

Zur Interpretation der Terminal-Funktionen wird nach Ausführung dieser Funktion die Terminal-Tabelle Nr. 4 benutzt. Diese Tabelle befindet sich im EPROM. Sie ist allerdings nur mit 0FFH vorbesetzt und kann vom Anwender für eine gewünschte Terminal-Emulation nachprogrammiert werden.

Funktion 47: **TERMINAL 3 AKTIVIEREN**

Zur Interpretation der Terminal-Funktionen wird nach Ausführung dieser Funktion die Terminal-Tabelle Nr. 3 benutzt. Diese Tabelle befindet sich im EPROM. Sie ist mit der info-S-VIDEO 7-Version der ADM 3A-Emulation belegt worden.

Funktion 48: **TERMINAL 2 AKTIVIEREN**

Zur Interpretation der Terminal-Funktionen wird nach Ausführung dieser Funktion die Terminal-Tabelle Nr. 2 benutzt. Diese Tabelle befindet sich im EPROM. Sie ist mit der Emulation eines GEVEKE VISA 30/40 belegt worden.

Funktion 49: **TERMINAL 1 AKTIVIEREN**

Zur Interpretation der Terminal-Funktionen wird nach Ausführung dieser Funktion die Terminal-Tabelle Nr. 1 benutzt. Diese Tabelle befindet sich im EPROM. Sie ist mit der Emulation eines ADDS VIEWPOINT belegt worden.

Funktion 50: **24 ZEILEN MIT 132 ZEICHEN PRO ZEILE**
(132 characters per line, 24 lines per screen)

VIDEO I geht auf die Darstellungsart 24 Zeilen mit 132 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Funktion 51: **24 ZEILEN MIT 96 ZEICHEN PRO ZEILE**
(96 characters per line, 24 lines per screen)

VIDEO I geht auf die Darstellungsart 24 Zeilen mit 96 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Funktion 52: **24 ZEILEN MIT 80 ZEICHEN PRO ZEILE**
(80 characters per line, 24 lines per screen)

VIDEO I geht auf die Darstellungsart 24 Zeilen mit 80 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Funktion 53: **US - ZEICHENSATZ** (US character set)

Für alle nachfolgenden Zeichen wird der amerikanische Zeichensatz vereinbart. Bereits von VIDEO I empfangene Zeichen werden von dieser Umschaltung nicht berührt.

Folgende ASCII-Codewort <--> Zeichenzuordnung gilt:
040H : Klammeraffe
05BH : Eckige Klammer auf
05CH : Schrägstrich rückwärts
05DH : Eckige Klammer zu
07BH : Geschweifte Klammer auf
07CH : Senkrechter Strich
07DH : Geschweifte Klammer zu
07EH : Schlange (Tilde)

Funktion 56: **TASTATURCODE ÄNDERN**
(change character code set -- keyboard)

Nach Übergabe des Kenncodes folgt das Zeichen, welches umcodiert werden soll. Das hierauf folgende Zeichen ersetzt dann für alle danach erfolgenden Tastatur-Eingaben das zuvor übergebene. Beispielsweise hat die Übertragung von:

Kenncode 'Y' 'Z'

zur Folge, daß bei jeder Betätigung der 'Y'-Taste ein 'Z' an den Computer gesendet wird. Bei Betätigung der 'Z'-Taste wird allerdings ebenfalls ein 'Z' an den Computer gesendet. Ein 'Y' steht nun nicht mehr zur Verfügung. Falls z.B. die Belegungen der 'Y'-Taste und der 'Z'-Taste vollständig getauscht werden sollen, ist die Übertragung folgender Sequenz notwendig:

```
Kenncode  'Y'  'Z'
Kenncode  'y'  'z'
Kenncode  19H  1AH  (Control-Belegung)
Kenncode  1AH  19H  (Control-Belegung)
Kenncode  'Z'  'Y'
Kenncode  'z'  'y'
```

Beispiel für 3 weitere Anwendungen:

Die Funktionstaste F1 mit dem Code 80H kann nicht mit einem String belegt werden, da 80H für interne Funktionen benutzt wird. Nach Übertragung der Sequenz:

Kenncode 80H 90H

hat die Funktionstaste F1 den Code 90H und kann mit einem String belegt werden.

Falls auf der Tastatur eine Taste mit dem Code 00H vorhanden ist, so kann diese nicht mit einem String belegt werden, da 00H kein gesetztes 8. Bit hat. Nach Übertragung der Sequenz:

Kenncode 00H 91H

hat diese Taste den Code 91H und kann mit einem String belegt werden.

Auf einer PC-Tastatur befindet sich eine Taste mit der Bezeichnung 'Pg Up'. Diese Taste hat den Code 89H. Falls mit dieser Taste die interne Funktion 'page up' ausgelöst werden soll, so ist das nach Übertragung der Sequenz:

Kenncode 89H 81H

möglich.

Funktion 57: **SEITENVORSCHUB MIT LÖSCHEN DES BILDSCHIRMES**
(formfeed and clear screen)

VIDEO I positioniert den CURSOR an den Anfang der nächsten Bildschirmseite und löscht anschließend den Inhalt dieser Bildschirmseite. Nach Ausführung der Funktion steht der Cursor in Zeile 0 und Spalte 0 (home position) des Bildschirmes. Der vorherige Bildschirminhalt kann durch 'Zurückblättern' (Funktion 32) wieder sichtbar gemacht werden.

Funktion 58: **ZEICHENFOLGEN LÖSCHEN** (clear string buffer)

Alle mit Funktion 59 gespeicherten Zeichenfolgen werden gelöscht. Der Zeichenpuffer steht erneut zur Verfügung.

Funktion 59: **ZEICHENFOLGEN LADEN** (load string)

Mit Hilfe dieser Funktion kann der Anwender eigene Funktionstasten definieren.

Mit VIDEO I können beliebige Zeichenfolgen gespeichert und auf Abruf durch die Tastatur an den Computer übergeben werden. Der hierzu reservierte Speicherplatz zur Aufnahme aller Zeichenfolgen beträgt 2KByte und ist so groß gewählt, daß ein Überlauf praktisch nicht vorkommt. Für den Aufruf einer gespeicherten Folge von der Tastatur aus sind alle Codeworte von 80H bis 0FFH zulässig.

Reihenfolge der einzugebenen Daten:
1. Kenncode zu Funktion 59
2. Zeichen, mit dem von der Tastatur später die Zeichenfolge aufgerufen wird
3. zu speichernde Zeichenfolge
4. Eingabe 00H zur Kennzeichnung des Endes der Zeichenfolge

Es ist zu beachten, daß das unter 2. eingegebene Zeichen im genannten Wertebereich von 80H bis 0FFH liegt, da anderenfalls die Funktion abgebrochen wird.

Funktion 60: **IDENTIFIKATION**

VIDEO I identifiziert sich mit seiner Versionsnummer. Von VIDEO I werden dazu 5 Bytes an den Computer übertragen:

Byte 1..4: Versionsnummer, ASCII-Ziffern z.B. '1', '1', '0', '2' falls es sich um die Version 1.102 handelt

Byte 5: Checksum über die Versionsnummer. Die Checksum wird wie folgt berechnet: Byte1 + Byte2 + Byte3 + Byte4 + 1 (8bit-Addition ohne Carry)

Diese Funktion wird zur Identifikation von VIDEO I durch das Betriebssystem benutzt.

Funktion 61: **CHARACTER HALBE HELLIGKEIT, BLINKEN UND UNTERSTREICHEN**
(underline + half intensity + blink)

Alle nachfolgenden Zeichen werden mit diesen 3 Attributen versehen. Falls vorher noch andere Attribute aktiv waren, so werden sie gleichzeitig deaktiviert. Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion keinen Einfluß.

Funktion 62: **CHARACTER BLINKEN UND UNTERSTREICHEN**
(underline + blink)

Alle nachfolgenden Zeichen werden mit diesen 2 Attributen versehen. Falls vorher noch andere Attribute aktiv waren, so werden sie gleichzeitig deaktiviert. Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion keinen Einfluß.

Funktion 63: **CHARACTER HALBE HELLIGKEIT UND UNTERSTREICHEN**
(half intensity + underline)

Alle nachfolgenden Zeichen werden mit diesen 2 Attributen versehen. Falls vorher noch andere Attribute aktiv waren, so werden sie gleichzeitig deaktiviert. Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion keinen Einfluß.

Funktion 64: **CHARACTER HALBE HELLIGKEIT, BLINKEN UND INVERS**
(single invers + half intensity + blink)

Alle nachfolgenden Zeichen werden mit diesen 3 Attributen versehen. Falls vorher noch andere Attribute aktiv waren, so werden sie gleichzeitig deaktiviert. Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion keinen Einfluß.

Funktion 65: **CHARACTER BLINKEN UND INVERS**
(single invers + blink)

Alle nachfolgenden Zeichen werden mit diesen 2 Attributen versehen. Falls vorher noch andere Attribute aktiv waren, so werden sie gleichzeitig deaktiviert. Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion keinen Einfluß.

Funktion 66: **CHARACTER HALBE HELLIGKEIT UND INVERS**
(single invers + half intensity)

Alle nachfolgenden Zeichen werden mit diesen 2 Attributen versehen. Falls vorher noch andere Attribute aktiv waren, so werden sie gleichzeitig deaktiviert. Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion keinen Einfluß.

Funktion 67: **CHARACTER HALBE HELLIGKEIT UND BLINKEN**
(half intensity blink)

Alle nachfolgenden Zeichen werden mit diesen 2 Attributen versehen. Falls vorher noch andere Attribute aktiv waren, so werden sie gleichzeitig deaktiviert. Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion keinen Einfluß.

Funktion 68: **CHARACTER NUR HALBE HELLIGKEIT** (half intensity only)

Alle nachfolgenden Zeichen werden mit diesem Attribut versehen. Falls vorher noch andere Attribute aktiv waren, so werden sie gleichzeitig deaktiviert. Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion keinen Einfluß.

Funktion 69: **CHARACTER NUR UNTERSTREICHEN** (underline only)

Alle nachfolgenden Zeichen werden mit diesem Attribut versehen. Falls vorher noch andere Attribute aktiv waren, so werden sie gleichzeitig deaktiviert. Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion keinen Einfluß.

Funktion 70: **CHARACTER NUR BLINKEN** (blink only)

Alle nachfolgenden Zeichen werden mit diesem Attribut versehen. Falls vorher noch andere Attribute aktiv waren, so werden sie gleichzeitig deaktiviert. Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion keinen Einfluß.

Funktion 71: **CHARACTER NUR INVERS** (single invers only)

Alle nachfolgenden Zeichen werden mit diesem Attribut versehen. Falls vorher noch andere Attribute aktiv waren, so werden sie gleichzeitig deaktiviert. Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion keinen Einfluß.

Funktion 72: **CHARACTER NORMAL** (normal video)

Alle gesetzten Attribute werden deaktiviert, so daß alle nachfolgenden Zeichen 'normal' dargestellt werden. Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion keinen Einfluß.

Funktion 73: **CHARACTER HALBE HELLIGKEIT AUS** (half video off)

Alle nachfolgenden Zeichen werden mit voller Intensität dargestellt.
Falls außerdem noch weitere Attribute aktiv sind, so werden sie von diesem Befehl nicht beeinflußt.
Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion ebenfalls keinen Einfluß.

Funktion 74: **CHARACTER HALBE HELLIGKEIT EIN** (half video on)

Alle nachfolgenden Zeichen werden mit halber Intensität dargestellt.
Falls außerdem noch weitere Attribute aktiv sind, so werden sie von diesem Befehl nicht beeinflußt.
Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion ebenfalls keinen Einfluß.

Funktion 75: **CHARACTER UNTERSTREICHEN AUS** (underline off)

Alle nachfolgenden Zeichen werden ohne Unterstreichung dargestellt.
Falls außerdem noch weitere Attribute aktiv sind, so werden sie von diesem Befehl nicht beeinflußt.
Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion ebenfalls keinen Einfluß.

Funktion 76: **CHARACTER UNTERSTREICHEN EIN** (underline on)

Alle nachfolgenden Zeichen werden unterstrichen dargestellt.
Falls außerdem noch weitere Attribute aktiv sind, so werden sie von diesem Befehl nicht beeinflußt.
Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion ebenfalls keinen Einfluß.

Funktion 77: **CHARACTER BLINKEN AUS** (blink off)

Alle nachfolgenden Zeichen werden nicht blinkend dargestellt.
Falls außerdem noch weitere Attribute aktiv sind, so werden sie von diesem Befehl nicht beeinflußt.
Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion ebenfalls keinen Einfluß.

Funktion 78: **CHARACTER BLINKEN EIN** (blink on)

Alle nachfolgenden Zeichen werden blinkend dargestellt.
Falls außerdem noch weitere Attribute aktiv sind, so werden sie von diesem Befehl nicht beeinflußt.
Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion ebenfalls keinen Einfluß.

Funktion 79: **CHARACTER INVERS AUS** (character invers off)

Alle nachfolgenden Zeichen werden nicht invers dargestellt. Invers ist in diesem Zusammenhang als invers zum Rest des Bildschirmes zu verstehen (d.h. schwarz auf weiß, falls der Bildschirm weiß auf schwarz arbeitet und weiß auf schwarz, falls der Bildschirm schwarz auf weiß arbeitet).

Falls außerdem noch weitere Attribute aktiv sind, so werden sie von diesem Befehl nicht beeinflußt.
Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion ebenfalls keinen Einfluß.

Funktion 80: **CHARACTER INVERS EIN** (single invers on)

Alle nachfolgenden Zeichen werden invers dargestellt. Invers ist in diesem Zusammenhang als invers zum Rest des Bildschirmes zu verstehen (d.h. schwarz auf weiß, falls der Bildschirm weiß auf schwarz arbeitet und weiß auf schwarz, falls der Bildschirm schwarz auf weiß arbeitet).
Falls außerdem noch weitere Attribute aktiv sind, so werden sie von diesem Befehl nicht beeinflußt.
Auf bereits zur VIDEO I übertragene Zeichen hat diese Funktion ebenfalls keinen Einfluß.

Funktion 81: **INVERSER BILDSCHIRM AUS**
(invers screen off)

Die gesamte Textfläche ist nicht invertiert, d.h. helle Schrift auf dunklem Hintergrund. Falls irgendwo im Text Funktion 80 'CHARACTER INVERS EIN' selektiert wird, erscheinen die Zeichen ab dort als dunkle Schrift auf hellem Grund.

Funktion 82: **INVERSER BILDSCHIRM EIN**
(invers screen on)

Die gesamte Textfläche ist invertiert, d.h. dunkle Schrift auf hellem Hintergrund. Falls irgendwo im Text Funktion 80 'CHARACTER INVERS EIN' selektiert wird, erscheinen die Zeichen ab dort als helle Schrift auf dunklem Grund.

Funktion 83: **TASTATUR AUSSCHALTEN** (keyboard lock)

Die Tastatur wird per Software abgeschaltet. Eingaben über die Tastatur bleiben ohne Wirkung mit Ausnahme von Funktion 84.

Funktion 84: **TASTATUR EINSCHALTEN** (keyboard unlock)

Erfolgt eine Eingabe des für diese Funktion vereinbarten Codes vom Computer oder von der Tastatur ist die Wirkung von Funktion 83 aufgehoben.

Funktion 85: **CURSOR AUS** (cursor off)

Der Cursor ist nicht mehr sichtbar. War der Cursor bereits vorher nicht sichtbar, ist der Aufruf dieser Funktion ohne Wirkung. Alle Funktionen, die sich auf die Cursorposition beziehen, bleiben erhalten, da der Cursor nur auf dem Bildschirm unsichtbar bleibt,

nicht aber für die interne Verarbeitung von VIDEO I.

Funktion 86: **CURSOR BLINKEN EIN** (cursor blink on)

Der Cursor erscheint als blinkender Cursor-Block auf dem Bildschirm. War der Cursor bereits vorher blinkend sichtbar, ist der Aufruf dieser Funktion ohne Wirkung.

Funktion 87: **CURSOR EIN** (cursor on)

Der Cursor erscheint nicht blinkend als Cursor-Block auf dem Bildschirm. War der Cursor bereits vorher nicht blinkend sichtbar, ist der Aufruf dieser Funktion ohne Wirkung.

Funktion 88: **KLINGEL** (bell)

Auf Steckerplatz S5 wird ein Impuls (negative Polarität bei Revision 1.0 und positive Polarität ab Revision 2.2 der VIDEO I-Hardware) zum Anschluß eines Summers ausgegeben. Bei Revision 1.0 der VIDEO I-Platine ist zur Ansteuerung dieses Summers eine Verlängerung der Impulsbreite durch ein Monoflop erforderlich. Ab Revision 2.2 der VIDEO I-Platine kann ein Piezo-Summer direkt angeschlossen werden.

Funktion 89: **TASTATUR-PUFFER DEAKTIVIEREN**
(keyboard buffer not activ)

Nach Aufruf dieser Funktion werden keine Zeichen von der Tastatur mehr zwischengespeichert. Ein eventuell zu diesem Zeitpunkt bereits gefüllter Tastatur-Puffer wird gleichzeitig gelöscht.

Jeder Tastendruck wird nun von VIDEO I sofort an der Computerschnittstelle bereitgestellt. Falls vom Computer das zuvor eingegebene Zeichen noch nicht von der VIDEO I abgefordert wurde, geht es verloren. Ausgenommen davon sind lediglich von der Tastatur ausgelöste Strings. Hier wird das jeweils nächste Zeichen des Strings erst dann an den Computer weitergegeben, wenn das zuvor bereitgestellte Zeichen bereits vom Computer abgefordert wurde.

Eine noch nicht beendete Stringausgabe wird von einem beliebigen Tastendruck sofort abgebrochen.

Funktion 90: **TASTATUR-PUFFER AKTIVIEREN**
(keyboard buffer activ)

Nach Aufruf dieser Funktion werden alle Zeichen von der Tastatur in der VIDEO I zwischengespeichert bis sie vom Computer benötigt werden. Als Zeichenpuffer steht ein Speicherbereich von 256Byte in der VIDEO I zur Verfügung. Bei Pufferüberlauf werden die letzten 256 Zeichen verworfen und der Puffer von neuem aufgefüllt.

Von der Tastatur ausgelöste Strings werden zuerst in den Tastaturpuffer kopiert, bevor sie an den Computer weitergereicht werden. Dabei ist zu beachten, daß ein String mit einer Länge von mehr als 256Byte regelmäßig einen Pufferüberlauf verursacht! Falls Funktionstasten mit derart langen Strings belegt werden sollen, ist es unbedingt erforderlich, den Tastatur-Puffer zu deaktivieren (Funktion 89).

Es ist zu beachten, daß auch mehrere hintereinander ausgelöste relativ lange Strings einen Pufferüberlauf verursachen können!

Funktion 91: **TASTATUR-PUFFER LÖSCHEN**
(clear keyboard buffer)

Nach Aufruf dieser Funktion ist der Tastatur-Puffer leer und wird von der VIDEO I erneut gefüllt, falls der Tastatur-Puffer aktiv ist. Falls der Tastatur-Puffer nicht aktiv ist, bleibt diese Funktion ohne Wirkung.

Funktion 92: **TEST-BILD AUSGEBEN** (display test pattern)

Der ganze Bildschirm wird mit dem Buchstaben 'H' gefüllt. Dieser Buchstabe eignet sich besonders gut zur Justage des Bildschirmes.

Diese Funktion ist nur in den Bildformaten 24/25 Zeilen mit 80/96 oder 132 Zeichen pro Zeile sinnvoll. In den anderen Bildformaten wird der Bildschirm mit einem Zufallsmuster gefüllt, daß von dem jeweiligen Zeichensatz abhängig ist.

Funktion 93: **NÄCHSTES ZEICHEN IGNORIEREN** (ignore next char)

Das nächste vom Computer an die VIDEO I übertragene Zeichen wird ignoriert. Diese Funktion ist zur 'Anpassung' von Terminal-Funktionen gedacht, die in dieser Form auf der VIDEO I nicht implementiert werden können (z.B. die Attribut-Vorbereitungsfunktionen des ADDS VIEWPOINT).

Funktion 94: **DIREKTES LADEN EINES 6845-REGISTERS**
(load 6845 direct)

Nach Übergabe des Kenncodes muß die Adresse des 6845-Registers (Wertebereich: 0 .. 15) und anschließend das in dieses Register einzuschreibende Byte übertragen werden. Dieses Byte wird sofort in das selektierte 6845-Register eingeschrieben. Eine Zwischenspeicherung erfolgt nicht. Bei Benutzung dieser Funktion ist zu beachten, daß einige VIDEO I-Funktionen ebenfalls Register des 6845 beschreiben und dadurch Werte, die mit Funktion 94 gesetzt wurden, in bestimmten Fällen wieder überschrieben werden können.

Diese Funktion ist in erster Linie für den erfahrenen Programmierer und Hardware-Fachmann gedacht, der einzelne Werte in den Registern des 6845-VIDEO-Controllers uminitialisieren möchte.

Anwendungsbeispiel: Nach Übertragung von

<ESC> 'X' 94 10 106

wird der Cursor als blinkender Unterstreichungsstrich dargestellt.

Funktion 95: **FENSTER-SCROLL NACH UNTEN (Y1,X1,Y2,X2)**
(window scroll down (y1,x1,y2,x2))

Der Text innerhalb des durch (y1,x1,y2,x2) beschriebenen Bildschirmfensters wird um eine Zeile nach unten verschoben. Die letzte Zeile des Bildschirmfensters wird dabei überschrieben. In der ersten Zeile des Bildschirmfensters wird eine Leerzeile eingefügt. Bei der Revision 1.0 der VIDEO I-Platine werden außerdem die Attribute "blinken", "invers" und "halbe Helligkeit" innerhalb des Fensters gelöscht. Ab Revision 2.0 der VIDEO I-Hardware werden diese Attribute mitverschoben.

(y1,x1) beschreibt den linken oberen Eckpunkt des Fensters in der Reihenfolge (Zeile, Spalte). (y2,x2) beschreibt den rechten unteren Eckpunkt des Fensters in der Reihenfolge (Zeile, Spalte). Die Eckpunkte werden als innerhalb des Fensters liegend interpretiert.

Die Numerierung beginnt mit Zeile 0 und Spalte 0. Dabei ist zu beachten, daß die X- und Y-Werte mit einem Offset von 32 = 20H (BLANK) erwartet werden.

Durch Aufruf der Sequenz

<ESC> 'X' 95 32 42 42 72

wird der Text in dem durch die Eckpunkte (Zeile 0, Spalte 10), (Zeile 10, Spalte 40) beschriebenen Fenster um eine Zeile nach unten verschoben.

Funktion 96: **FENSTER-SCROLL NACH OBEN (Y1,X1,Y2,X2)**
(window scroll up (y1,x1,y2,x2))

Der Text innerhalb des durch (y1,x1,y2,x2) beschriebenen Bildschirmfensters wird um eine Zeile nach oben verschoben. Die erste Zeile des Bildschirmfensters wird dabei überschrieben. In der letzten Zeile des Bildschirmfensters wird eine Leerzeile eingefügt. Bei der Revision 1.0 der VIDEO I-Platine werden außerdem die Attribute "blinken", "invers" und "halbe Helligkeit" innerhalb des Fensters gelöscht. Ab Revision 2.0 der VIDEO I-Hardware werden diese Attribute mitverschoben.

(y1,x1) beschreibt den linken oberen Eckpunkt des Fensters in der Reihenfolge (Zeile, Spalte). (y2,x2) beschreibt den rechten unteren Eckpunkt des Fensters in der Reihenfolge (Zeile, Spalte). Die Eckpunkte werden als innerhalb des Fensters liegend interpretiert.

Die Numerierung beginnt mit Zeile 0 und Spalte 0. Dabei ist zu beachten, daß die X- und Y-Werte mit einem Offset von 32 = 20H (BLANK) erwartet werden.

Durch Aufruf der Sequenz

<ESC> 'X' 96 32 42 42 72

wird der Text in dem durch die Eckpunkte (Zeile 0, Spalte 10), (Zeile 10, Spalte 40) beschriebenen Fenster um eine Zeile nach oben verschoben.

Funktion 97: **FENSTER-INHALT LÖSCHEN (Y1,X1,Y2,X2)**
(clear window (y1,x1,y2,x2))

Der Text innerhalb des durch (y1,x1,y2,x2) beschriebenen Bildschirmfensters wird gelöscht.

(y1,x1) beschreibt den linken oberen Eckpunkt des Fensters in der Reihenfolge (Zeile, Spalte). (y2,x2) beschreibt den rechten unteren Eckpunkt des Fensters in der Reihenfolge (Zeile, Spalte). Die Eckpunkte werden als innerhalb des Fensters liegend interpretiert.

Die Numerierung beginnt mit Zeile 0 und Spalte 0. Dabei ist zu beachten, daß die X- und Y-Werte mit einem Offset von 32 = 20H (BLANK) erwartet werden.

Durch Aufruf der Sequenz

<ESC> 'X' 97 32 42 42 72

wird der Text in dem durch die Eckpunkte (Zeile 0, Spalte 10), (Zeile 10, Spalte 40) beschriebenen Fenster gelöscht.

Funktion 98: **ZEICHENDARSTELLUNG AUF DEM BILDSCHIRM EINSCHALTEN**
(screen dark off)

Die mit Funktion 99 ausgeschaltete Zeichendarstellung auf dem Bildschirm wird wieder eingeschaltet. Dabei werden alle im Bildwiederholspeicher stehenden Zeichen mit ihren Attributen wieder angezeigt. Dies gilt auch für Zeichen, die während der ausgeschalteten Zeichendarstellung in den Bildschirm geschrieben wurden.

Funktion 99: **ZEICHENDARSTELLUNG AUF DEM BILDSCHIRM AUSSCHALTEN**
(screen dark on)

Der gesamte Bildschirm wird dunkel gesteuert (bzw. hell gesteuert, falls vorher mit Funktion 82 der Bildschirm auf 'INVERS' umgeschaltet wurde), ohne daß im Bildwiederholspeicher Zeichen gelöscht werden.

Nach Auslösen dieser Funktion findet zwar keine Bildschirmanzeige mehr statt, aber alle VIDEO I-Funktionen werden intern ohne irgendwelche Beeinträchtigungen ausgeführt.

Bei ausgeschalteter Zeichendarstellung auf dem Bildschirm werden die internen Funktionen der VIDEO I um ca. 20% schneller.

Diese Funktion kann z.B. genutzt werden um die Bildröhre eines angeschlossenen Monitors zu schonen, wenn keine Aktionen auf einem ansonsten bereiten Computer stattfinden (gilt nicht für inverse Darstellung).

In der Revision 1.0 der VIDEO I-Hardware ist das Ausschalten der Zeichendarstellung auf dem Bildschirm nicht möglich. Diese Funktion ist deshalb dort ohne Wirkung.

Funktion 100: **25 ZEILEN MIT 132 ZEICHEN PRO ZEILE**
(132 characters per line, 25 lines per screen)

VIDEO I geht auf die Darstellungsart 25 Zeilen mit 132 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Funktion 101: **25 ZEILEN MIT 96 ZEICHEN PRO ZEILE**
(96 characters per line, 25 lines per screen)

VIDEO I geht auf die Darstellungsart 25 Zeilen mit 96 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Funktion 102: **25 ZEILEN MIT 80 ZEICHEN PRO ZEILE**
(80 characters per line, 25 lines per screen)

VIDEO I geht auf die Darstellungsart 25 Zeilen mit 80 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Funktion 103: **24 ZEILEN MIT 66 ZEICHEN PRO ZEILE**
(66 characters per line, 24 lines per screen)

VIDEO I geht auf die Darstellungsart 24 Zeilen mit 66 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Das Attribut 'unterstrichene Zeichendarstellung' steht bei 66 Zeichen pro Zeile nicht mehr zur Verfügung.

Diese Funktion ist nur in den Versionen 3 und 5 der VIDEO I-Platine vorhanden. Bei den anderen Versionen erfolgt bei Auslösung dieser Funktion keine Aktion.

Die Versionen 3 und 5 werden durch eine 3 bzw. 5 als letzte Ziffer der Versionsnummer gekennzeichnet. Die Versionsnummer kann mit Funktion 60 von VIDEO I angefordert werden. Außerdem wird die Versionsnummer nach einem 'RESET' in der 1. Bildschirmzeile angezeigt.

Funktion 104: **24 ZEILEN MIT 48 ZEICHEN PRO ZEILE**
(48 characters per line, 24 lines per screen)

VIDEO I geht auf die Darstellungsart 24 Zeilen mit 48 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Das Attribut 'unterstrichene Zeichendarstellung' steht bei 48 Zeichen pro Zeile nicht mehr zur Verfügung.

Diese Funktion ist nur in den Versionen 3 und 5 der VIDEO I-Platine vorhanden. Bei den anderen Versionen erfolgt bei Auslösung dieser Funktion keine Aktion.

Die Versionen 3 und 5 werden durch eine 3 bzw. 5 als letzte Ziffer der Versionsnummer gekennzeichnet. Die Versionsnummer kann mit Funktion 60 von VIDEO I angefordert werden. Außerdem wird die Versionsnummer nach einem 'RESET' in der 1. Bildschirmzeile angezeigt.

Funktion 105: **24 ZEILEN MIT 40 ZEICHEN PRO ZEILE**
(40 characters per line, 24 lines per screen)

VIDEO I geht auf die Darstellungsart 24 Zeilen mit 40 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Das Attribut 'unterstrichene Zeichendarstellung' steht bei 40 Zeichen pro Zeile nicht mehr zur Verfügung.

Diese Funktion ist nur in den Versionen 3 und 5 der VIDEO I-Platine vorhanden. Bei den anderen Versionen erfolgt bei Auslösung dieser Funktion keine Aktion.

Die Versionen 3 und 5 werden durch eine 3 bzw. 5 als letzte Ziffer der Versionsnummer gekennzeichnet. Die Versionsnummer kann mit Funktion 60 von VIDEO I angefordert werden. Außerdem wird die Versionsnummer nach einem 'RESET' in der 1. Bildschirmzeile angezeigt.

Funktion 106: **12 ZEILEN MIT 132 ZEICHEN PRO ZEILE**
(132 characters per line, 12 lines per screen)

VIDEO I geht auf die Darstellungsart 12 Zeilen mit 132 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Das Attribut 'unterstrichene Zeichendarstellung' steht bei 12 Zeilen pro Bildschirmseite nicht mehr zur Verfügung.

Diese Funktion ist nur in den Versionen 4 und 5 der VIDEO I-Platine vorhanden. Bei den anderen Versionen erfolgt bei Auslösung dieser Funktion keine Aktion.

Die Versionen 4 und 5 werden durch eine 4 bzw. 5 als letzte Ziffer der Versionsnummer gekennzeichnet. Die Versionsnummer kann mit Funktion 60 von VIDEO I angefordert werden. Außerdem wird die Versionsnummer nach einem 'RESET' in der 1. Bildschirmzeile angezeigt.

Funktion 107: **12 ZEILEN MIT 96 ZEICHEN PRO ZEILE**
(96 characters per line, 12 lines per screen)

VIDEO I geht auf die Darstellungsart 12 Zeilen mit 96 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Das Attribut 'unterstrichene Zeichendarstellung' steht bei 12 Zeilen pro Bildschirmseite nicht mehr zur Verfügung.

Diese Funktion ist nur in den Versionen 4 und 5 der VIDEO I-Platine vorhanden. Bei den anderen Versionen erfolgt bei Auslösung dieser Funktion keine Aktion.

Die Versionen 4 und 5 werden durch eine 4 bzw. 5 als letzte Ziffer der Versionsnummer gekennzeichnet. Die Versionsnummer kann mit Funktion 60 von VIDEO I angefordert werden. Außerdem wird die Versionsnummer nach einem 'RESET' in der 1. Bildschirmzeile angezeigt.

Funktion 108: **12 ZEILEN MIT 80 ZEICHEN PRO ZEILE**
(80 characters per line, 12 lines per screen)

VIDEO I geht auf die Darstellungsart 12 Zeilen mit 80 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Das Attribut 'unterstrichene Zeichendarstellung' steht bei 12 Zeilen pro Bildschirmseite nicht mehr zur Verfügung.

Diese Funktion ist nur in den Versionen 4 und 5 der VIDEO I-Platine vorhanden. Bei den anderen Versionen erfolgt bei Auslösung dieser Funktion keine Aktion.

Die Versionen 4 und 5 werden durch eine 4 bzw. 5 als letzte Ziffer der Versionsnummer gekennzeichnet. Die Versionsnummer kann mit Funktion 60 von VIDEO I angefordert werden. Außerdem wird die Versionsnummer nach einem 'RESET' in der 1. Bildschirmzeile angezeigt.

Funktion 109: **12 ZEILEN MIT 66 ZEICHEN PRO ZEILE**
(66 characters per line, 12 lines per screen)

VIDEO I geht auf die Darstellungsart 12 Zeilen mit 66 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Das Attribut 'unterstrichene Zeichendarstellung' steht bei 66 Zeichen pro Zeile nicht mehr zur Verfügung.

Diese Funktion ist nur in der Version 5 der VIDEO I-Platine vorhanden. Bei den anderen Versionen erfolgt bei Auslösung dieser Funktion keine Aktion.

Die Version 5 wird durch eine 5 als letzte Ziffer der Versionsnummer gekennzeichnet. Die Versionsnummer kann mit Funktion 60 von VIDEO I angefordert werden. Außerdem wird die Versionsnummer nach einem 'RESET' in der 1. Bildschirmzeile angezeigt.

Funktion 110: **12 ZEILEN MIT 48 ZEICHEN PRO ZEILE**
(48 characters per line, 12 lines per screen)

VIDEO I geht auf die Darstellungsart 12 Zeilen mit 48 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Das Attribut 'unterstrichene Zeichendarstellung' steht bei 48 Zeichen pro Zeile nicht mehr zur Verfügung.

Diese Funktion ist nur in der Version 5 der VIDEO I-Platine vorhanden. Bei den anderen Versionen erfolgt bei Auslösung dieser Funktion keine Aktion.

Die Version 5 wird durch eine 5 als letzte Ziffer der Versionsnummer gekennzeichnet. Die Versionsnummer kann mit Funktion 60 von VIDEO I angefordert werden. Außerdem wird die Versionsnummer nach einem 'RESET' in der 1. Bildschirmzeile angezeigt.

Funktion 111: **12 ZEILEN MIT 40 ZEICHEN PRO ZEILE**
(40 characters per line, 12 lines per screen)

VIDEO I geht auf die Darstellungsart 12 Zeilen mit 40 Zeichen über. Gleichzeitig wird der gesamte Bildwiederholspeicher gelöscht und der Cursor in Zeile 0 und Spalte 0 (home position) positioniert. Alle anderen voreingestellten Werte bleiben erhalten.

Das Attribut 'unterstrichene Zeichendarstellung' steht bei 40 Zeichen pro Zeile nicht mehr zur Verfügung.

Diese Funktion ist nur in der Version 5 der VIDEO I-Platine vorhanden. Bei den anderen Versionen erfolgt bei Auslösung dieser Funktion keine Aktion.

Die Version 5 wird durch eine 5 als letzte Ziffer der Versionsnummer gekennzeichnet. Die Versionsnummer kann mit Funktion 60 von VIDEO I angefordert werden. Außerdem wird die Versionsnummer nach einem 'RESET' in der 1. Bildschirmzeile angezeigt.

Funktion 112: **PUSH CURSOR**

Die augenblickliche absolute Cursorposition wird in den internen Cursor-Stack 'gepusht', d.h. auf der aktuellen Position des internen Kellerspeichers für Cursor-Positionen abgelegt. Gleichzeitig wird der interne Zeiger (Stackpointer) auf die nächste freie Position des Kellerspeichers weitergeschaltet.

Der Cursor-Stack hat eine Tiefe von 1000 Cursor-Positionen, d.h., der Kellerspeicher kann 1000 Cursor-Positionen aufnehmen.

Funktion 113: **POP CURSOR**

Der interne Zeiger (Stackpointer) wird um eine Position im internen Kellerspeicher zurückgeschaltet. Anschließend wird der Cursor auf die an dieser Stelle im Kellerspeicher abgelegte Cursorposition gesetzt.

Der Cursor steht nach Ausführung dieser Funktion wieder an der gleichen Stelle im 16KByte großen Bildwiederholspeicher wie zu dem Zeitpunkt als die Cursorposition im Kellerspeicher abgelegt wurde. Falls in der Zwischenzeit der Anfang des sichtbaren Bildschirmes verschoben wurde (z.B. durch 'scroll up' oder 'scroll down') hat sich diese Cursorposition relativ mit verschoben. Absolut steht der Cursor aber auf demselben Character wie vorher.

Falls das Bild in der Zwischenzeit vollständig 'herausgescrollt' wurde, steht der Cursor nun auf einer unsichtbaren aber wohldefinierten Position.

Diese Tatsache kann z.B. wie folgt genutzt werden:

- Funktion 37 auslösen (erase screen and cursor home)
- Funktion 112 auslösen (push cursor)
- Funktion 57 auslösen (formfeed)
- Funktion 113 auslösen (pop cursor)

Der Cursor ist nun unsichtbar und steht wohldefiniert in Zeile 0 und Spalte 0 (home position) der vorhergehenden Bildschirmseite. Es ist nun möglich in diese vorgehende Seite zunächst unsichtbare Informationen einzuschreiben. Durch Auslösen der Funktion 112 kann dann der (immer noch unsichtbare) Cursor erneut im Kellerspeicher abgelegt werden.

Mit einer absoluten Cursorpositionierung (z.B. Funktion 9) wird der Cursor wieder auf dem Bildschirm sichtbar und es kann nun ab dieser Position wieder in den sichtbaren Teil des Bildwiederholspeichers geschrieben werden.

Die zunächst unsichtbaren Informationen in der vorhergehenden Seite können jederzeit durch 'Zurückblättern' (Funktion 32) zur Anzeige gebracht werden.

Funktion 114: **SWAP CURSOR**

Analog zur Funktion 113 wird der Cursor auf die an unterster Stelle im Kellerspeicher abgelegte Cursorposition gesetzt. Gleichzeitig wird die vorherige aktuelle Cursorposition analog zur Funktion 112 im Kellerspeicher abgelegt.

Die augenblickliche Cursorposition und die an tiefster Position im Kellerspeicher befindliche Cursorposition tauschen also ihren Platz.

Diese Funktion eignet sich besonders gut um zwischen 2 Cursorpositionen hin- und herzuspringen.

Funktion 115: **RESET CURSOR STACK**

Der Cursor-Stackpointer wird auf den Anfang des Kellerspeichers für Cursor-Positionen (Cursor-Stack) zurückgesetzt.

Es steht nun wieder die volle Tiefe von 1000 Positionen im Cursor-Stack zur Verfügung.

7. Erstellen eines neuen Alfa-Zeichensatzes

Die Code-Matrix des Zeichensatzes ist im EPROM auf der VIDEO I abgelegt. Die Größe eines Zeichensatzes ist abhängig von dem jeweiligen Bildschirmformat. Je nach Bildformat, Firmware-Version und EPROM-Version können gleichzeitig 4 bis 16 verschiedene Zeichensätze abgelegt werden. Jeweils die Hälfte davon werden bei 132 Zeichen pro Zeile und bei 80/96 Zeichen pro Zeile benutzt.

Das Zeichensatz-EPROM **1.002** enthält 4 Zeichensätze (2 für 80/96 und 2 für 132 Zeichen pro Zeile). Der jeweils 2. Zeichensatz ist mit den unterstrichenen Zeichen des 1. Zeichensatzes belegt. Der jeweilige 2. Zeichensatz wird mit der Funktion 'Unterstreichen einschalten' aktiviert.

Das Zeichensatz-EPROM **1.003** enthält 6 Zeichensätze. Zusätzlich zu den 4 Zeichensätzen des EPROMs 1.002 enthält dieses EPROM noch die Zeichensätze für 24x40/48 Zeichen und 24x66 Zeichen pro Bildschirmseite.

Das Zeichensatz-EPROM **1.004** enthält ebenfalls 6 Zeichensätze. Zusätzlich zu den 4 Zeichensätzen des EPROMs 1.002 enthält dieses EPROM noch die Zeichensätze für 12x80/96 Zeichen und 12x132 Zeichen pro Bildschirmseite.

Das Zeichensatz-EPROM **1.005** enthält 10 Zeichensätze. Zusätzlich zu den 6 Zeichensätzen des EPROMs 1.003 sind die beiden zusätzlichen Zeichensätze des EPROMs 1.004 und die Zeichensätze für 12x40/48 und 12x66 Zeichen pro Bildschirmseite enthalten.

Zum Erstellen eines neuen Zeichensatzes sind Änderungen im Zeichensatz-EPROM notwendig.

7.1 Codierung der Zeichensätze im EPROM

Im 80/96-Mode mit 24 oder 25 Zeilen pro Bildschirmseite wird jedes Zeichen auf dem Bildschirm in einer 8x11-Punktmatrix dargestellt. Bei 24 Zeilen zu 96 Zeichen pro Zeile werden somit 264 Bildschirmzeilen (Punktzeilen) zu 768 Punkten je Zeile angezeigt.

Im 132-Mode mit 24 oder 25 Zeilen pro Bildschirmseite wird jedes Zeichen auf dem Bildschirm in einer 6x11-Punktmatrix dargestellt. Bei 24 Zeilen zu 132 Zeichen pro Zeile werden somit 264 Bildschirmzeilen zu 792 Punkten je Zeile angezeigt.

Im 80/96-Mode mit 12 Zeilen pro Bildschirmseite wird jedes Zeichen auf dem Bildschirm in einer 8x22-Punktmatrix dargestellt. Bei 12 Zeilen zu 96 Zeichen pro Zeile werden somit 264 Bildschirmzeilen (Punktzeilen) zu 768 Punkten je Zeile angezeigt.

Im 132-Mode mit 12 Zeilen pro Bildschirmseite wird jedes Zeichen auf dem Bildschirm in einer 6x22-Punktmatrix dargestellt. Bei 12 Zeilen zu 132 Zeichen pro Zeile werden somit 264 Bildschirmzeilen zu 792 Punkten je Zeile angezeigt.

Im 40/48-Mode mit 24 Zeilen pro Bildschirmseite wird jedes Zeichen auf dem Bildschirm in einer 16x11-Punktmatrix dargestellt. Bei 24 Zeilen zu 48 Zeichen pro Zeile werden somit 264 Bildschirmzeilen (Punktzeilen) zu 768 Punkten je Zeile angezeigt.

Im 66-Mode mit 24 Zeilen pro Bildschirmseite wird jedes Zeichen auf dem Bildschirm in einer 12x11-Punktmatrix dargestellt. Bei 24 Zeilen zu 66 Zeichen pro Zeile werden somit 264 Bildschirmzeilen zu 792 Punkten je Zeile angezeigt.

Im 40/48-Mode mit 12 Zeilen pro Bildschirmseite wird jedes Zeichen auf dem Bildschirm in einer 16x22-Punktmatrix dargestellt. Bei 12 Zeilen zu 48 Zeichen pro Zeile werden somit 264 Bildschirmzeilen (Punktzeilen) zu 768 Punkten je Zeile angezeigt.

Im 66-Mode mit 12 Zeilen pro Bildschirmseite wird jedes Zeichen auf dem Bildschirm in einer 12x22-Punktmatrix dargestellt. Bei 12 Zeilen zu 66 Zeichen pro Zeile werden somit 264 Bildschirmzeilen zu 792 Punkten je Zeile angezeigt.

Zwischenräume zwischen den Buchstaben und Zeilen müssen in allen Fällen in der Punktmatrix eines jeden Buchstaben berücksichtigt werden.

7.1.1 Codierung des Character-EPROMs 1.002

Das Character-EPROM 1.002 enthält 4 Tabellen für die 4 implementierten Zeichensätze. Diese 4 Tabellen befinden sich auf folgenden Adressen:

```
0000H bis 07FFH: Zeichensatz 1 (24/25x80/96)                Z96.DAT
0800H bis 0FFFH: Zeichensatz 2 (24/25x80/96 'underline')    Z96.DAT
1000H bis 17FFH: Zeichensatz 1 (24/25x132)                 Z132.DAT
1800H bis 1FFFH: Zeichensatz 2 (24/25x132   'underline')   Z132.DAT
```

In jedem Zeichensatz sind 128 verschiedene Zeichen darstellbar. Für jedes dieser Zeichen sind zur Codierung 16Byte im Zeichensatz erforderlich.

Die ersten 11 Byte enthalten die Punktcodierung des darzustellenden Zeichens für die 11 Zeilen dieses Zeichens in folgender Form:

Das 1. Byte enthält die Codierung für die 1. darzustellende Punktzeile, das 2. Byte für die 2. darzustellende Zeile und das 11. Byte schließlich für die 11. darzustellende Zeile. Für jeden Punkt, der auf dem Bildschirm erleuchtet dargestellt werden soll, ist eine binäre 1 und für jeden Punkt, der dunkel erscheinen soll, eine binäre 0 programmiert. Dabei ist jedoch zu beachten, daß das niederwertigste Bit dem linken äußeren Punkt und das höchstwertigste Bit dem rechten äußeren Punkt zugeordnet ist.

Bit 0 (also der äußerste linke Punkt) ist im Zeichensatz 1.002 in der Regel 0, um einen Zwischenraum zwischen den einzelnen Buchstaben zu erreichen. Eine Ausnahme gilt nur für Zeichen, die lückenlos aneinandergereiht werden sollen (z.B. Grafikzeichen). Das 1.

Byte eines jeden Zeichens ist ebenfalls in der Regel 0, um einen Zwischenraum zwischen den einzelnen Zeilen zu realisieren. Das 11. Byte ist im jeweils 2. Zeichensatz mit dem Unterstreichungsstrich belegt worden. Im 132-Zeichen-Mode wird Bit 6 und 7 nicht angezeigt, da zur Darstellung eines Zeichens in der Zeile nur 6 Bit benutzt werden.

Beispiel: Das Zeichen 'R' soll mit der folgenden Punktmatrix dargestellt werden:

```
#####
#    #
#    #
#####
#  #
#   #
#    #
```

Daraus ergibt sich folgende Verschlüsselung:

```
Byte  1: 00000000          Leerzeile
Byte  2: 00111110          #####     (rückwärts)
Byte  3: 01000010          #    #    (rückwärts)
Byte  4: 01000010          #    #    (rückwärts)
Byte  5: 00111110          #####     (rückwärts)
Byte  6: 00010010          #  #      (rückwärts)
Byte  7: 00100010          #   #     (rückwärts)
Byte  8: 01000010          #    #    (rückwärts)
Byte  9: 00000000          leer      (Platz für Unterlängen)
Byte 10: 00000000          leer      (Platz für Unterlängen)
Byte 11: 00000000          leer      (Platz für Unterstreichung)
```

Das 12. Byte muß immer mit 0FFH belegt werden. Es wird zur Darstellung nicht benutzt.

Die Bytes 13 bis 16 müssen beim jeweils 1. Zeichensatz für jedes Zeichen mit dem Code des Zeichens selbst belegt sein. Bei dem Zeichen 'R' z.B. müssen die Bytes 13 bis 16 mit 052H belegt sein.

Beim jeweils 2. Zeichensatz müssen die Bytes 13 bis 16 mit dem Code des Zeichens selbst plus gesetztem 8. Bit belegt sein. Bei dem Zeichen 'R' z.B. sind diese Bytes mit 0D2H belegt.

Alle 128 darstellbaren Zeichen sind hintereinander im Zeichensatz programmiert. Das Zeichen 'R' (mit dem Code 052H) belegt somit die Adressen 0520H bis 052FH im Zeichensatz.

Im Anhang ist der Zeichensatz 1.002 abgedruckt worden. Dabei ist zu beachten, daß jeweils 2 Tabellen hintereinander gelistet wurden. Z96.DAT beinhaltet die beiden Zeichensätze für 80/96 Zeichen pro Zeile und Z132.DAT die beiden Zeichensätze für 132 Zeichen pro Zeile.

7.1.2 Codierung des Character-EPROMs 1.003

Das Character-EPROM 1.003 enthält die 4 Tabellen des Character-EPROMs 1.002 sowie die 2 zusätzlichen Tabellen für 24 Zeilen mit 40/48 Zeichen pro Zeile und 24 Zeilen mit 66 Zeichen pro Zeile. Diese 6 Tabellen befinden sich auf folgenden Adressen:

```
0000H bis 0FFFH: Zeichensatz für 24x40/48                    Z48.DAT
1000H bis 1FFFH: Zeichensatz für 24x66                       Z66.DAT
2000H bis 27FFH: Zeichensatz 1 (24/25x80/96)                 Z96.DAT
2800H bis 2FFFH: Zeichensatz 2 (24/25x80/96 'underline')     Z96.DAT
3000H bis 37FFH: Zeichensatz 1 (24/25x132)                  Z132.DAT
3800H bis 3FFFH: Zeichensatz 2 (24/25x132   'underline')   Z132.DAT
```

Die Codierung der Zeichensätze Z96 und Z132 kann dem Kapitel 7.1.1 entnommen werden. In diesem Kapitel wird nur die Codierung der Zeichensätze Z48 und Z66 erläutert.

In beiden Zeichensätzen sind 128 verschiedene Zeichen darstellbar. Für jedes dieser Zeichen sind zur Codierung 32Byte im Zeichensatz erforderlich.

Jedes einzelne Zeichen wird in 2 Hälften zu je 16Byte codiert. Die ersten 2KByte der beiden Zeichensätze enthalten die linke Hälfte und die zweiten 2KByte die rechte Hälfte eines Zeichens. Für beide 'Halbzeichen' gelten dann die Codierungsregeln des Kapitel 7.1.1, d.h. Z48 ist analog zu Z96 und Z66 analog zu Z132 zu verstehen!

Beispiel: Das Zeichen 'R' soll mit der folgenden Punktmatrix dargestellt werden:

```
##########
##        ##
##        ##
##########
##    ##
##      ##
##        ##
```

Daraus ergibt sich folgende Verschlüsselung:

1. Hälfte (linker Teil)

```
Byte  1: 00000000          Leerzeile
Byte  2: 11111100          #######   (rückwärts)
Byte  3: 00001100          ##        (rückwärts)
Byte  4: 00001100          ##        (rückwärts)
Byte  5: 11111100          #######   (rückwärts)
Byte  6: 00001100          ##        (rückwärts)
Byte  7: 00001100          ##        (rückwärts)
Byte  8: 00001100          ##        (rückwärts)
Byte  9: 00000000          leer      (Platz für Unterlängen)
Byte 10: 00000000          leer      (Platz für Unterlängen)
Byte 11: 00000000          Leerzeile
```

2. Hälfte (rechter Teil)

```
Byte  1: 00000000          Leerzeile
Byte  2: 00001111        ####        (rückwärts)
Byte  3: 00110000            ##      (rückwärts)
Byte  4: 00110000            ##      (rückwärts)
Byte  5: 00001111        ####        (rückwärts)
Byte  6: 00000011        ##          (rückwärts)
Byte  7: 00001100          ##        (rückwärts)
Byte  8: 00110000            ##      (rückwärts)
Byte  9: 00000000          leer      (Platz für Unterlängen)
Byte 10: 00000000          leer      (Platz für Unterlängen)
Byte 11: 00000000          Leerzeile
```

Für beide Hälften gilt:

Das 12. Byte muß immer mit 0FFH belegt werden. Es wird zur Darstellung nicht benutzt. Die Bytes 13 bis 16 müssen bei der linken Hälfte für jedes Zeichen mit dem Code des Zeichens selbst belegt sein. Bei der rechten Hälfte ist für die Bytes 13 bis 16 zusätzlich das 8. Bit zu setzen. Bei dem Zeichen 'R' z.B. müssen die Bytes 13 bis 16 in der linken Hälfte mit 052H und in der rechten Hälfte mit 0D2H belegt sein.

Alle 128 darstellbaren Zeichen sind in 2 Hälften hintereinander im Zeichensatz programmiert. Das Zeichen 'R' (mit dem Code 052H) belegt somit die Adressen 0520H bis 052FH (1. Hälfte) und 0D20H bis 0D2FH (2. Hälfte) im Zeichensatz.

7.1.3 Codierung des Character-EPROMs 1.004

Das Character-EPROM 1.004 enthält die 4 Tabellen des Character-EPROMs 1.002 sowie die 2 zusätzlichen Tabellen für 12 Zeilen mit 80/96 Zeichen pro Zeile und 12 Zeilen mit 132 Zeichen pro Zeile. Diese 6 Tabellen befinden sich auf folgenden Adressen:

```
0000H bis 0FFFH: Zeichensatz für 12x80/96                  Z1296.DAT
1000H bis 1FFFH: Zeichensatz für 12x132                   Z12132.DAT
2000H bis 27FFH: Zeichensatz 1 (24/25x80/96)                 Z96.DAT
2800H bis 2FFFH: Zeichensatz 2 (24/25x80/96 'underline')     Z96.DAT
3000H bis 37FFH: Zeichensatz 1 (24/25x132)                  Z132.DAT
3800H bis 3FFFH: Zeichensatz 2 (24/25x132   'underline')    Z132.DAT
```

Die Codierung der Zeichensätze Z96 und Z132 kann dem Kapitel 7.1.1 entnommen werden. In diesem Kapitel wird nur die Codierung der Zeichensätze Z1296 und Z12132 erläutert.

In beiden Zeichensätzen sind 128 verschiedene Zeichen darstellbar. Für jedes dieser Zeichen sind zur Codierung 32Byte im Zeichensatz erforderlich.

Jedes einzelne Zeichen wird in 2 Hälften zu je 16Byte codiert. Die ersten 2KByte der beiden Zeichensätze enthalten die untere Hälfte und die zweiten 2KByte die obere Hälfte eines Zeichens. Für beide 'Halbzeichen' gelten dann die Codierungsregeln des Kapitel 7.1.1,

d.h., Z1296 ist analog zu Z96 und Z12132 analog zu Z132 zu verstehen!

Beispiel: Das Zeichen 'R' soll mit der folgenden Punktmatrix dargestellt werden:

```
#####
#####
#    #
#    #
#    #
#    #
#####
#####
#  #
#  #
#   #
#   #
#    #
#    #
```

Daraus ergibt sich folgende Verschlüsselung:

1. Hälfte (unterer Teil)

Byte 1: 00010010	# #	(rückwärts)
Byte 2: 00100010	# #	(rückwärts)
Byte 3: 00100010	# #	(rückwärts)
Byte 4: 01000010	# #	(rückwärts)
Byte 5: 01000010	# #	(rückwärts)
Byte 6: 00000000	leer	(Platz für Unterlängen)
Byte 7: 00000000	leer	(Platz für Unterlängen)
Byte 8: 00000000	leer	(Platz für Unterlängen)
Byte 9: 00000000	leer	(Platz für Unterlängen)
Byte 10: 00000000	Leerzeile	
Byte 11: 00000000	Leerzeile	

2. Hälfte (oberer Teil)

Byte 1: 00000000	Leerzeile	
Byte 2: 00000000	Leerzeile	
Byte 3: 00111110	#####	(rückwärts)
Byte 4: 00111110	#####	(rückwärts)
Byte 5: 01000010	# #	(rückwärts)
Byte 6: 01000010	# #	(rückwärts)
Byte 7: 01000010	# #	(rückwärts)
Byte 8: 01000010	# #	(rückwärts)
Byte 9: 00111110	#####	(rückwärts)
Byte 10: 00111110	#####	(rückwärts)
Byte 11: 00010010	# #	(rückwärts)

Für beide Hälften gilt:

Das 12. Byte muß immer mit 0FFH belegt werden. Es wird zur Darstellung nicht benutzt. Die Bytes 13 bis 16 müssen bei der unteren Hälfte für jedes Zeichen mit dem Code des Zeichens selbst belegt sein. Bei der oberen Hälfte ist für die Bytes 13 bis 16 zusätzlich das 8. Bit zu setzen. Bei dem Zeichen 'R' z.B. müssen die Bytes 13 bis 16 in der unteren Hälfte mit 052H und in der oberen Hälfte mit 0D2H belegt sein.

Alle 128 darstellbaren Zeichen sind in 2 Hälften hintereinander im Zeichensatz programmiert. Das Zeichen 'R' (mit dem Code 052H) belegt somit die Adressen 0520H bis 052FH (1. Hälfte) und 0D20H bis 0D2FH (2. Hälfte) im Zeichensatz.

7.1.4 Codierung des Character-EPROMs 1.005

Das Character-EPROM 1.005 enthält die 6 Tabellen des Character-EPROMs 1.003, die 2 zusätzlichen Tabellen für 12 Zeilen mit 80/96 Zeichen pro Zeile und 12 Zeilen mit 132 Zeichen pro Zeile sowie die 2 zusätzlichen Tabellen für 12 Zeilen mit 40/48 Zeichen pro Zeile und 12 Zeilen mit 66 Zeichen pro Zeile. Diese 10 Tabellen befinden sich auf folgenden Adressen:

0000H bis 0FFFH: Zeichensatz für 12x40/48	Z1248.DAT
1000H bis 1FFFH: Zeichensatz für 12x66	Z1266.DAT
2000H bis 2FFFH: Zeichensatz für 12x80/96	Z1296.DAT
3000H bis 3FFFH: Zeichensatz für 12x132	Z12132.DAT
4000H bis 4FFFH: Zeichensatz für 24x40/48	Z48.DAT
5000H bis 5FFFH: Zeichensatz für 24x66	Z66.DAT
6000H bis 67FFH: Zeichensatz 1 (24/25x80/96)	Z96.DAT
6800H bis 6FFFH: Zeichensatz 2 (24/25x80/96 'underline')	Z96.DAT
7000H bis 77FFH: Zeichensatz 1 (24/25x132)	Z132.DAT
7800H bis 7FFFH: Zeichensatz 2 (24/25x132 'underline')	Z132.DAT

Die Codierung der Zeichensätze Z1296, Z12132, Z48, Z66, Z96 und Z132 kann den Kapiteln 7.1.1 bis 7.1.3 entnommen werden. In diesem Kapitel wird nur die Codierung der Zeichensätze Z1248 und Z1266 erläutert.

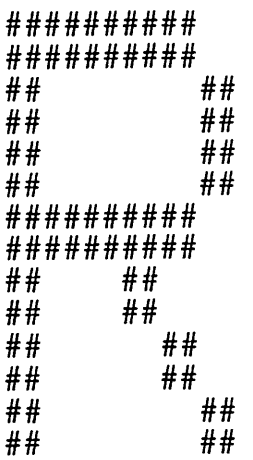
In beiden Zeichensätzen sind 128 verschiedene Zeichen darstellbar. Für jedes dieser Zeichen sind zur Codierung 64Byte im Zeichensatz erforderlich.

Für 128 Zeichen sind somit 8KByte für einen Zeichensatz erforderlich. Da jedoch nur ein Platz von 4KByte zur Verfügung steht erfolgt die Abspeicherung in einer komprimierten Form. Bevor der Zeichensatz komprimiert wird, ist zuerst einmal eine Version in nicht komprimierter Form zu erstellen.

Jedes einzelne Zeichen wird dazu in 4 Teilen zu je 16Byte codiert. Die ersten 2KByte der beiden Zeichensätze enthalten das linke untere Viertel, die zweiten 2KByte das linke obere Viertel, die dritten 2KByte das rechte obere Viertel und die vierten 2KByte das rechte untere Viertel eines Zeichens. Für jedes 'Viertelzeichen'

gelten dann die Codierungsregeln des Kapitels 7.1.1, d.h. Z1248 ist analog zu Z96 und Z1266 analog zu Z132 zu verstehen!

Beispiel: Das Zeichen 'R' soll mit der folgenden Punktmatrix dargestellt werden:



Daraus ergibt sich folgende Verschlüsselung:

1. Viertel (linker unterer Teil)

Byte 1: 00001100	##	(rückwärts)
Byte 2: 00001100	##	(rückwärts)
Byte 3: 00001100	##	(rückwärts)
Byte 4: 00001100	##	(rückwärts)
Byte 5: 00001100	##	(rückwärts)
Byte 6: 00000000	leer	(Platz für Unterlängen)
Byte 7: 00000000	leer	(Platz für Unterlängen)
Byte 8: 00000000	leer	(Platz für Unterlängen)
Byte 9: 00000000	leer	(Platz für Unterlängen)
Byte 10: 00000000	Leerzeile	
Byte 11: 00000000	Leerzeile	

2. Viertel (linker oberer Teil)

Byte 1: 00000000	Leerzeile	
Byte 2: 00000000	Leerzeile	
Byte 3: 11111100	######	(rückwärts)
Byte 4: 11111100	######	(rückwärts)
Byte 5: 00001100	##	(rückwärts)
Byte 6: 00001100	##	(rückwärts)
Byte 7: 00001100	##	(rückwärts)
Byte 8: 00001100	##	(rückwärts)
Byte 9: 11111100	######	(rückwärts)
Byte 10: 11111100	######	(rückwärts)
Byte 11: 00001100	##	(rückwärts)

3. Viertel (rechter oberer Teil)

```
Byte  1: 00000000          Leerzeile
Byte  2: 00000000          Leerzeile
Byte  3: 00001111        ####        (rückwärts)
Byte  4: 00001111        ####        (rückwärts)
Byte  5: 00110000            ##      (rückwärts)
Byte  6: 00110000            ##      (rückwärts)
Byte  7: 00110000            ##      (rückwärts)
Byte  8: 00110000            ##      (rückwärts)
Byte  9: 00001111        ####        (rückwärts)
Byte 10: 00001111        ####        (rückwärts)
Byte 11: 00000011        ##          (rückwärts)
```

4. Viertel (rechter unterer Teil)

```
Byte  1: 00000011        ##          (rückwärts)
Byte  2: 00001100          ##        (rückwärts)
Byte  3: 00001100          ##        (rückwärts)
Byte  4: 00110000            ##      (rückwärts)
Byte  5: 00110000            ##      (rückwärts)
Byte  6: 00000000          leer      (Platz für Unterlängen)
Byte  7: 00000000          leer      (Platz für Unterlängen)
Byte  8: 00000000          leer      (Platz für Unterlängen)
Byte  9: 00000000          leer      (Platz für Unterlängen)
Byte 10: 00000000          Leerzeile
Byte 11: 00000000          Leerzeile
```

Für jedes Viertel gilt:

Das 12. Byte muß immer mit 0FFH belegt werden. Es wird zur Darstellung nicht benutzt. Die Bytes 13 bis 16 bleiben zunächst undefiniert.

Alle 128 darstellbaren Zeichen sind in 4 Teilen hintereinander im Zeichensatz programmiert. Das Zeichen 'R' (mit dem Code 052H) belegt somit die Adressen 0520H bis 052FH (1. Viertel), 0D20H bis 0D2FH (2. Viertel), 1520H bis 152FH (3. Viertel) und 1D20H bis 1D2FH (4. Viertel) im Zeichensatz.

Um nun eine Abspeicherung in nur 4KBytes zu ermöglichen ist der nach obigen Regeln erstellte Zeichensatz auf doppelte Viertel zu durchsuchen. So ist z.B. das linke obere Viertel des Buchstabens 'R' auch bei den Buchstaben 'B', 'E', 'F' und 'P' vorhanden. Es wird deshalb nur einmal abgespeichert.

Wenn der Zeichensatz nach obiger Komprimierung immer noch größer als 4KByte ist, muß auf einen Teil der Zeichen (z.B. die schrägen Pfeile) verzichtet werden. Diese Zeichen sind dann durch Blank zu ersetzen.

Nach dem Komprimieren auf 4KBytes sind noch folgende Punkte zu beachten:

- Das Leerzeichen muß auf die Adressen 0200H bis 020FH plaziert werden (Blank besteht aus 4 gleichen Vierteln!). Die Zuordnung aller anderen Viertelzeichen ist willkürlich.

- Für jedes Viertelzeichen sind nun die Bytes 13 bis 16 mit ihrer Adresse dividiert durch 16 (ohne Berücksichtigung des Restes) zu belegen. Für das Byte auf der Adresse 020EH ist somit eine 20H vorzusehen.

- Als drittes ist eine Tabelle mit Zeigern auf den komprimierten Zeichensatz zu erstellen. Für jedes Zeichen sind in dieser Tabelle 4 Bytes einzutragen. Jedes Byte multipliziert mit 16 ist ein Zeiger auf ein Viertelzeichen im komprimierten Zeichensatz.
  Das 1. Byte zeigt auf das linke untere Viertel, das 2. Byte auf das linke obere Viertel, das 3. Byte auf das rechte obere Viertel und das 4. Byte auf das rechte untere Viertel eines Zeichens.

  Die so konstruierte Tabelle hat eine Länge von 512Bytes und muß im Programm-EPROM der VIDEO I enthalten sein. Die Tabelle für den Zeichensatz Z1248 (ZTAB48.DAT) belegt die Adressen 2000H bis 21FFH und die Tabelle für den Zeichensatz Z1266 (ZTAB66.DAT) die Adressen 2200H bis 23FFH im Programm-EPROM 'VIDEO 1.105'.

7.2 Hilfsprogramme zur Erstellung neuer Alfa-Zeichensätze

Zur Erstellung neuer Alfa-Zeichensätze existieren die Hilfsprogramme CHAR, CHRBREIT, CHRHOCH, CHRHB, CHRDEL und CHRCOMP.

Diese Hilfsprogramme sind auf der Diskette 'Hilfsprogramme zur VIDEO I' als ausführbare Programme (ablauffähig unter CP/M 2.2, HKM-ZDOS und CP/M PLUS) vorhanden. Das Programm 'CHAR' ist zusätzlich als Quellfile (assemblierbar mit M80 von Microsoft) auf der Diskette 'Quelltext der VIDEO I-Firmware' enthalten.

Die Dateien Z96.DAT, Z132.DAT, Z48.DAT, Z66.DAT, Z1296.DAT, Z12132.DAT, Z1248.DAT und Z1266.DAT sind ebenfalls auf der Diskette mit den Hilfsprogrammen enthalten.

7.2.1 Erstellung der Zeichensätze 'Z96' und 'Z132' mit dem Dienstprogramm 'CHAR'

Um die Erstellung eines neuen Zeichensatzes für 80/96 Zeichen pro Zeile und 132 Zeichen pro Zeile zu erleichtern, existiert das Dienstprogramm 'CHAR', mit dessen Hilfe auch der im Anhang abgebildete Zeichensatz ausgedruckt wurde.

Außerdem können mit 'CHAR' auch die Halb- bzw. Viertelzeichen der anderen Zeichensätze erstellt bzw. verschönert werden.

Die Zeichensätze für 80/96 Zeichen pro Zeile (Z96.DAT) und für 132 Zeichen pro Zeile (Z132.DAT) werden mit 'CHAR' getrennt erstellt. Beide erzeugten Tabellen müssen anschließend hintereinander in ein EPROM programmiert werden.

Die Erstellung eines Zeichensatzes erfolgt interaktiv am Bildschirm durch Zeichnen der Punktmatrix für den Buchstaben. Dabei sind Korrekturen jederzeit möglich. Der Cursor kann hierbei in der gleichen Art bewegt werden, wie man es von dem Textverarbeitungsprogramm WordStar von MicroPro gewohnt ist. Die Cursor-Steuerung ist auf die VIDEO I abgestimmt.

Aufruf des Programmes 'CHAR':

CHAR 'name'

'name' ist der Name einer alten Zeichensatz-Datei, die bearbeitet werden soll. 'CHAR' erzeugt einen neuen Zeichengenerator grundsätzlich mit dem Namen 'CHARGEN.DAT'. Dieser Name kann bei Bedarf im Betriebssystem umbenannt werden.

Nach dem Aufruf erfragt 'CHAR' als erstes den Code (die 'Adresse') des zu bearbeitenden Zeichens in Hex. Danach wird das Zeichen auf dem Bildschirm dargestellt und kann bearbeitet werden.

Byte 12 wird von 'CHAR' automatisch mit 0FFH belegt. Die Bytes 13 bis 16 werden von 'CHAR' automatisch mit der 'Adresse' des Zeichens vorbesetzt.

Während der Bearbeitung eines Zeichens sind folgende Befehle möglich:

^X: Mit dem Cursor eine Zeile nach unten gehen
^E: Mit dem Cursor eine Zeile nach oben gehen
^S: Mit dem Cursor einen Punkt nach links gehen
^A: Mit dem Cursor an den linken Rand des Zeichens gehen
^D: Mit dem Cursor einen Punkt nach rechts gehen
^F: Mit dem Cursor an den rechten Rand des Zeichens gehen
X: Setzen eines Punktes an der Cursor-Position
Blank: Löschen eines Punktes an der Cursor-Position
^T: Löschen der Zeile, in der sich der Cursor befindet
^O: Das ganze Zeichen wird gelöscht.
ESC: 'CHAR' fragt nach einem neuen Code, der bearbeitet werden soll. An dieser Stelle kann mit ^C in das Betriebssystem zurückgesprungen werden. Dabei ist aber zu beachten, daß der augenblicklich bearbeitete Zeichensatz hier nicht zurückgeschrieben wird! Es muß vorher bei der Bearbeitung eines Zeichens der Code ^V eingegeben werden!
CR: 'CHAR' geht zu dem nächsten Zeichen weiter.
LF: 'CHAR' geht zu dem vorherigen Zeichen zurück.
^V: Zeichengenerator abspeichern
^P: Zeichengenerator auf dem Drucker ausdrucken
^G: Zeichen kopieren; 'CHAR' fragt nach dem Code des Zeichens, von dem die Punktmatrix in das gerade bearbeitete Zeichen kopiert werden soll.

'^' bedeutet, daß gleichzeitig mit dem Buchstaben die Control-Taste gedrückt werden muß.

7.2.2 Erstellung der Zeichensätze 'Z48' und 'Z66' mit dem Dienstprogramm 'CHRBREIT'

'CHRBREIT' ist ein Dienstprogramm zur Erstellung der Breitschrift-Zeichensätze 'Z48.DAT' und 'Z66.DAT'. Diese beiden Zeichensätze werden durch Verbreitern der vorhandenen Zeichensätze 'Z96.DAT' und 'Z132.DAT' gewonnen.

Aufruf des Programmes 'CHRBREIT':

CHRBREIT

Es müssen sich die Dateien 'Z96.DAT' und 'Z132.DAT' (mit diesen Namen!) auf der gleichen Diskette befinden. 'CHRBREIT' erzeugt dann die Dateien 'Z48.TMP' und 'Z66.TMP'. Diese Dateien können dann bei Bedarf im Betriebssystem umbenannt werden (z.B. in 'Z48.DAT' und 'Z66.DAT').

Mit Hilfe des Programmes 'CHAR' können nun noch Verschönerungen an den Halbzeichen der obigen Zeichensätze vorgenommen werden.

7.2.3 Erstellung der Zeichensätze 'Z1296' und 'Z12132' mit dem Dienstprogramm 'CHRHOCH'

'CHRHOCH' ist ein Dienstprogramm zur Erstellung der Hochschrift-Zeichensätze 'Z1296.DAT' und 'Z12132.DAT'. Diese beiden Zeichensätze werden durch Erhöhen der vorhandenen Zeichensätze 'Z96.DAT' und 'Z132.DAT' gewonnen.

Aufruf des Programmes 'CHRHOCH':

CHRHOCH

Es müssen sich die Dateien 'Z96.DAT' und 'Z132.DAT' (mit diesen Namen!) auf der gleichen Diskette befinden. 'CHRHOCH' erzeugt dann die Dateien 'Z1296.TMP' und 'Z12132.TMP'. Diese Dateien können dann bei Bedarf im Betriebssystem umbenannt werden (z.B. in 'Z1296.DAT' und 'Z12132.DAT').

Mit Hilfe des Programmes 'CHAR' können nun noch Verschönerungen an den Halbzeichen der obigen Zeichensätze vorgenommen werden.

7.2.4 Erstellung der Zeichensätze 'Z1248' und 'Z1266' mit den Dienstprogrammen 'CHRHB', 'CHRDEL' und 'CHRCOMP'

'CHRHB' ist ein Dienstprogramm zur Erstellung der nicht komprimierten Hoch/Breit-Zeichensätze 'Z1248L.DAT' und 'Z1266L.DAT'. Diese beiden Zeichensätze werden durch Erhöhen der vorhandenen Zeichensätze 'Z48.DAT' und 'Z66.DAT' gewonnen.

Aufruf des Programmes 'CHRHB':

CHRHB

Es müssen sich die Dateien 'Z48.DAT' und 'Z66.DAT' (mit diesen Namen!) auf der gleichen Diskette befinden. 'CHRHB' erzeugt dann die Dateien 'Z1248L.TMP' und 'Z1266L.TMP'. Diese Dateien können dann bei Bedarf im Betriebssystem umbenannt werden (z.B. in 'Z1248L.DAT' und 'Z1266L.DAT').

Die Dateien 'Z1248L.TMP' und 'Z1266L.TMP' haben jeweils eine Länge von 8KBytes und müssen zur Verwendung erst komprimiert werden.

'CHRCOMP' ist ein Dienstprogramm zur Erstellung der komprimierten Hoch/Breit-Zeichensätze 'Z1248.DAT' und 'Z1266.DAT'. Diese beiden Zeichensätze werden durch komprimieren der vorhandenen Zeichensätze 'Z1248L.TMP' und 'Z1266L.TMP' gewonnen.

Aufruf des Programmes 'CHRCOMP':

CHRCOMP

Es müssen sich die Dateien 'Z1248L.TMP' und 'Z1266L.TMP' (mit diesen Namen!) auf der gleichen Diskette befinden. 'CHRCOMP' erzeugt dann die Dateien 'Z1248.TMP' und 'Z1266.TMP'. Gleichzeitig

werden die beiden Tabellen 'ZTAB48.TMP' und 'ZTAB66.TMP' erzeugt. Diese Dateien können dann bei Bedarf im Betriebssystem umbenannt werden (z.B. in 'Z1248.DAT', 'Z1266.DAT', 'ZTAB48.DAT' und 'ZTAB66.DAT').

Mit Hilfe des Programmes 'CHAR' können nun noch Verschönerungen an den Viertelzeichen der obigen komprimierten Zeichensätze vorgenommen werden.

Falls sich die Zeichensätze Z1248L oder Z1266L nicht auf 4KByte komprimieren lassen, erfolgt von 'CHRCOMP' eine diesbezügliche Fehlermeldung. Aus der Angabe der letzten komprimierten Zeichenposition kann dann entnommen werden, wieviele Zeichen in dem betreffenden Zeichensatz durch Leerzeichen ersetzt werden müssen.

Beispiel einer Fehlermeldung von 'CHRCOMP':

```
ZEICHENSATZ "Z1248L" KANN NICHT KOMPRIMIERT WERDEN
LETZTE KOMPRIMIERTE ZEICHENPOSITION:    124
```

Die Zeichen auf den Positionen 125, 126, 127 und 128 (also insgesamt 4 Zeichen) konnten nicht mehr komprimiert werden.

In diesem Beispiel sollten nun 4 Zeichen im Zeichensatz 'Z1248L' durch Leerzeichen ersetzt werden.

'CHRDEL' ist ein Dienstprogramm zum Löschen von einzelnen Zeichen in den nicht komprimierten Hoch/Breit-Zeichensätzen 'Z1248L.DAT' und 'Z1266L.DAT'.

Aufruf des Programmes 'CHRDEL':

CHRDEL 'name1' 'name2'

'name1' ist der Name der Datei, aus der Zeichen gelöscht werden sollen. Diese Datei muß ein nicht komprimierter Hoch/Breit-Zeichensatz von 8KByte Länge sein. Als Typbezeichnung im Filenamen wird 'DAT' festgelegt. Typische Bezeichnungen für 'name1' sind 'Z1248L' oder 'Z1266L'.

'name2' ist eine Datei mit den Codes der Zeichen, die aus dem Zeichensatz 'name1' gelöscht werden sollen. Für jedes zu löschende Zeichen ist der Code in dezimaler Schreibweise mit 3 Dezimalziffern in der Datei 'name2' enthalten. Für jedes Zeichen ist eine neue Zeile zu beginnen. Als Typbezeichnung im Filenamen wird auch hier 'DAT' festgelegt.

Den Bezeichnungen 'name1' und 'name2' kann jeweils noch eine Laufwerksbezeichnung vorangestellt sein.

'CHRDEL' erzeugt als Ausgabe eine neue Datei 'name1.TMP' auf dem gleichen Laufwerk, auf dem 'name1.DAT' gelesen wurde. 'name1.TMP' kann dann direkt als Eingabe für 'CHRCOMP' benutzt werden.

Beispiel:

Im Zeichensatz 'Z1248L' sollen die schrägen Pfeile gelöscht werden. Dazu wird mit einem Editor eine Datei 'DEL48.DAT' mit folgendem Inhalt erstellt (jeweils in der 1. Spalte einer neuen Zeile begonnen):

005
006
007
008

Durch Aufruf von

CHRDEL Z1248L DEL48

wird eine neue Datei 'Z1248L.TMP' erzeugt, in der die schrägen Pfeile durch Leerzeichen ersetzt sind.

7.3 Erstellung der Zeichensatz-EPROMs aus den verschiedenen Zeichensätzen

Die verschiedenen Zeichensatz-EPROMs werden durch Zusammenkopieren der jeweiligen Zeichensätze erstellt:

Zeichensatz-EPROM 1.002:

PIP CHR1.002=Z96.DAT,Z132.DATÄOÜ

Zum Programmieren ist ein EPROM vom Typ 2764 notwendig.

Zeichensatz-EPROM 1.003:

PIP CHR1.003=Z48.DAT,Z66.DAT,Z96.DAT,Z132.DATÄOÜ

Zum Programmieren ist ein EPROM vom Typ 27128 notwendig.

Zeichensatz-EPROM 1.004:

PIP CHR1.004=Z1296.DAT,Z12132.DAT,Z96.DAT,Z132.DATÄOÜ

Zum Programmieren ist ein EPROM vom Typ 27128 notwendig.

Zeichensatz-EPROM 1.005:

PIP CHR1.005=Z1248.DAT,Z1266.DAT,Z1296.DAT,Z12132.DAT,CHR1.003ÄOÜ

Zum Programmieren ist ein EPROM vom Typ 27256 notwendig.

Zusätzlich müssen beim Zeichensatz-EPROM 1.005 noch die Tabellen 'ZTAB48.DAT' und 'ZTAB66.DAT' in das Programm-EPROM eingetragen werden.

Dazu ist zuerst das Programm-EPROM (Version 5 erforderlich!) auf eine Diskette zu kopieren. Anschließend werden die Adressen 2000H bis 23FFH mit den beiden obigen Tabellen überschrieben.

Dies kann unter CP/M PLUS mit dem SID erfolgen. Dazu sind folgende Eingaben notwendig:

```
SID
Rprogrammname
RZTAB48.DAT 2000
RZTAB66.DAT 2200
Wprogrammname
^C
```

Unter HKM-ZDOS kann dazu der PSP benutzt werden. Die Eingaben sehen dann wie folgt aus:

```
PSP
Iprogrammname
R
IZTAB48.DAT
R2100
IZTAB66.DAT
R2300
Iprogrammname
W100,40FF
^C
```

8. Erstellen und Laden eines Text-Zeichensatzes

Im Grafik-Modus gibt es für die Darstellung von Buchstaben und Ziffern den Text-Modus. In dieser Betriebsart werden besondere ladbare Zeichensätze benutzt, die im Alfa-Modus nicht zur Verfügung stehen. Umgekehrt stehen die Zeichensätze des Alfa-Modus im Text-Modus ebenfalls nicht zur Verfügung.

Die VIDEO I kann in der Standardversion 4 ladbare Zeichensätze verwalten. In den Versionen 3, 4 und 5 stehen darüber hinaus noch 2 weitere fest installierte Zeichensätze im EPROM zur Verfügung.

8.1 Codierung der Text-Zeichensätze

Jeweils zwei Zeichensätze belegen einen durchgehenden Speicherbereich von 4096 Byte (4KByte).

Die 4096 Byte teilen sich auf nach:

2816 Byte für Zeichensatz 1 (bzw. 3 oder 5)
1280 Byte für Zeichensatz 2 (bzw. 4 oder 6).

Jedes darstellbare Zeichen setzt sich aus einer Folge von n Byte zusammen. Jedes Byte besitzt m gültige Bit. Ein Zeichen wird demnach durch eine n mal m Punktmatrix dargestellt. Für die Parameter n und m gelten die Grenzwerte:

Zeichenhöhe	n <= 255
Zeichenbreite	m <= 8.

Im Text-Modus unterscheidet die VIDEO I 128 darstellbare Zeichen. Für jedes darzustellende Zeichen sind n Bytes mit m gültigen Bits hintereinander angeordnet. Alle darzustellenden Zeichen folgen in einem Zeichensatz unmittelbar hintereinander. Für die Bitreihenfolge gilt das gleiche, wie bei einem Alfa-Zeichensatz.

Das Bild des ersten Zeichens enthält zwei Byte für die Beschreibung der Struktur des Zeichensatzes, und geht so als darstellbares Zeichen verloren. Das 1. Byte enthält den Wert n und das 2. Byte den Wert m.

Die Datenmenge für einen Zeichensatz umfaßt n mal 128 Byte. n ist so zu wählen, daß der verfügbare Speicherraum nicht überschritten wird. Der erste Zeichensatz eines 4096Byte-Blockes kann bei Bedarf den zweiten, der dann nicht selektiert werden darf, überschreiben.

Beispiel:

Als Beispiel für die Struktur des 4096Byte-Blockes seien hier Zeichensatz 5 und 6 beschrieben, die in der erweiterten Version der VIDEO I im Programm-EPROM enthalten sind.

Zeichensatz 5 belegt 2560 Byte. Jedes Zeichen wird durch 20 Byte dargestellt; jedes Byte enthält 8 gültige Bit, d.h.

n = 20,
m = 8.

Die Zahlen n und m belegen die ersten beiden Byte des ersten Zeichens. Dieses Zeichen kann zwar, sollte jedoch nicht auf dem Bildschirm dargestellt werden.

256 Byte hinter Zeichensatz 5 sind nicht belegt.

Zeichensatz 6 belegt die restlichen 1280 Byte des 4096Byte-Blockes. Jedes Zeichen wird mit 10 Byte dargestellt. Die Anzahl der gültigen Bit pro Byte beträgt 8. Es gilt:

n = 10
m = 8

Die Steuerbyte n und m sind wieder im ersten Zeichen eingelagert.

Die Zeichen des Zeichensatzes 5 sind doppelt so hoch wie die Zeichen des Zeichensatzes 6. Auf dem Bildschirm dargestellt ergeben sich gleiche Buchstabenhöhen, wenn im Grafik-Modus 1 der Zeichensatz 5 und im Grafik-Modus 2 der Zeichensatz 6 verwendet wird.

Die Zeichensätze 5 und 6 sind in den Versionen 3, 4 und 5 des Programm-EPROMs ab Adresse 3000H enthalten.

8.2 Hilfsprogramme zur Erstellung neuer Text-Zeichensätze

Das Programm TRCHAR

TRCHAR transformiert einen Zeichensatz, der mit dem Programm CHAR erzeugt wurde, in die Form des Zeichensatzes für den Text-Modus.

TRCHAR verarbeitet nur die ersten 128 Zeichen und nicht die unterstrichenen Zeichen. Das Programm erzeugt eine Datei, die den transformierten Zeichensatz zweimal enthält. Die erste Form besteht aus Zeichen, die doppelt so hoch sind, wie die Zeichen der zweiten Form. Jedes Zeichen wird mit 20 bzw. 10 Byte dargestellt und nicht mit 11 wie im Original.

Aufruf:

TRCHAR quelldatei zieldatei bits

mit quelldatei	Name der Quelldatei (z.B. Z96.DAT oder Z132.DAT)
zieldatei	Name der Zieldatei; eine existierende Datei mit demselben Namen wird überschrieben
bits	Anzahl der gültigen Bit pro Byte (1..8)

Wird das Programm ohne Parameter gestartet, so fragt es nach den fehlenden Werten.

Ein mit 'TRCHAR' erzeugter Zeichensatz kann mit dem Programm 'LDCHAR' in die VIDEO I geladen werden. Ein solcher Zeichensatz kann aber auch im Programm-EPROM ab Adresse 3000H abgelegt werden.

Das Programm LDCHAR

LDCHAR lädt den mit TRCHAR erzeugten Zeichensatz in die VIDEO I.

Aufruf:

LDCHAR zdatei zeichensatz

mit zdatei Name der Datei, die Zeichensatz enthält

zeichensatz Kennbuchstabe ('B' oder 'C') für das Ziel
'B' steht für Zeichensatz 1 und 2 laden
'C' steht für Zeichensatz 3 und 4 laden

Wird das Programm ohne Parameter gestartet, so fragt es nach den fehlenden Werten.

9. Der Kommandointerpreter 'ALFA'

'ALFA' ist ein Kommandointerpreter zur Auslösung aller im Kapitel 6 dieses Handbuches definierten ALFA-Funktionen der VIDEO I. Zusätzlich werden von 'ALFA' noch einige zusätzliche Kommandos ausgeführt.

'ALFA' ist besonders gut geeignet für:

- Demonstrationen der ALFA-Funktionen in der VIDEO I
- Interaktive Einstellung von ALFA-Funktionen in der VIDEO I
- Automatische Einstellungen von ALFA-Funktionen der VIDEO I

Mit Hilfe von 'ALFA' können z.B. Bildschirmformate und Stringbelegung einer Tastatur beim 'Hochfahren' eines Betriebssystems (unter CP/M PLUS --> PROFILE.SUB, unter HKM-ZDOS --> INIT.DO) eingestellt werden.

'ALFA' kann auf zwei verschiedene Arten aufgerufen werden:

- ALFA
- ALFA filename

In der ersten Form werden alle Kommandos von der augenblicklichen Konsole angefordert. 'ALFA' meldet sich mit '*' als Aufforderung zur Eingabe eines Kommandos.

In der zweiten Form enthält die Datei filename alle Kommandos, die von 'ALFA' ausgeführt werden sollen. Falls für filename kein Typ angegeben wird, ist 'DEM' als Standardtyp vorgesehen.

In beiden Fällen erfolgen die Ausgaben von 'ALFA' direkt auf die VIDEO I unter Umgehung des jeweiligen Betriebssystems. Die Kommandos für 'ALFA' sind ebenfalls in beiden Fällen gleich.

9.1 Syntax der einzelnen 'ALFA'-Kommandos

Alle Kommandos sind nach folgender Syntax aufgebaut:

name n1 n2 n3 n4 text

Die einzelnen Parameter einer Eingabezeile werden durch mindestens ein Leerzeichen voneinander getrennt. Ein Parameter kann niemals mit einem Leerzeichen beginnen. Eine Eingabezeile darf maximal aus 110 Zeichen bestehen. Ein einzelner Parameter kann maximal 96 Zeichen lang sein.

name ist der Name eines auszuführenden Kommandos. name muß immer in der ersten Spalte einer Zeile beginnen.
Falls in der ersten Spalte einer Zeile ein ';' steht, wird diese Zeile als Kommentar betrachtet und nicht interpretiert.
In einer Kommandozeile kann name auch fehlen. In diesem Fall beginnt die Kommandozeile mit einem führenden Leerzeichen in der ersten Spalte.

Groß- und Kleinschreibung wird bei einem Kommando nicht unterschieden.

n1..n4 sind numerische Parameter. Diese können dezimal, hexadezimal oder als Literale geschrieben werden.
Eine Dezimalzahl enthält nur die Ziffern 0..9 und optional ein nachfolgendes 'D' oder 'd'. Eine Hexadezimalzahl enthält nur die Ziffern 0..9, die Buchstaben a..f oder die Buchstaben A..F und ein nachfolgendes 'H' oder 'h'. Ein Literal ist hier ein Zeichen zwischen '!' und 'ß' (einschließlich), daß zwischen zwei Apostrophe gesetzt wird (Achtung: ' ' ist hier nicht erlaubt!).
Es können bis zu vier numerische Parameter in einem Kommando vorkommen. n1..n4 können aber auch ganz fehlen.

text ist ein beliebiger ASCII-Text. text kann auch fehlen.

Falls mit n1..n4 Zeilen- und Spalten-Positionen auf dem Bildschirm beschrieben werden, so ist zu beachten, daß hier immer mit EINS beginnend gezählt wird! Die HOME-Position befindet sich hier also in Zeile 1 und Spalte 1!

9.2 Alle 'ALFA'-Kommandos in alfabetischer Reihenfolge

Jedes unbekannte Kommando hat eine Ausgabe von <CR> und <LF> auf der VIDEO I zur Folge. Eine weitergehende Fehlermeldung erfolgt nicht!

BLINK

Das Attribut 'Blinken' wird eingeschaltet (--> Funktion 78).

CBLINK

'Cursor blinkend ein' (--> Funktion 86)

CHANGEK n1 n2

Das Zeichen n1 wird auf der Tastatur (Keyboard) durch das Zeichen n2 ersetzt (--> Funktion 56).

CHANGES n1 n2

Das Zeichen n1 wird auf dem Bildschirm (Screen) durch das Zeichen n2 ersetzt (--> Funktion 55).

CDOWN

Cursor Down (--> Funktion 17)

CLEFT

Cursor Left (--> Funktion 19)

CLENDL

Clear End of Line (--> Funktion 35)

CLENDS

Clear End of Screen (--> Funktion 40)

CLL

Clear Line (--> Funktion 34)

CLS

Clear Screen (--> Funktion 37)

CLSTARTS

Clear Start of Screen (--> Funktion 39)

CLSTRBUF

Clear String Buffer (--> Funktion 58)

COFF

'Cursor off' (--> Funktion 85)

CONSOLE

Alle weiteren Kommandos werden von der Konsole erwartet.

CPOP

'POP Cursor' (--> Funktion 113)

CPUSH

'PUSH Cursor' (--> Funktion 112)

CR

Carriage Return (--> Funktion 6)

CRIGHT

Cursor Right (--> Funktion 20)

CSTADY

'Cursor nicht blinkend (stady) ein' (--> Funktion 87)

CSTRES

'Reset Cursor Stack' (--> Funktion 115)

CSWAP

'SWAP Cursor' (--> Funktion 114)

CUP

Cursor Up (--> Funktion 18)

CURSOR n1 n2

Cursor in Zeile n1 und Spalte n2 positionieren (--> Funktion 10)

DARKOFF

Dark Off (--> Funktion 98)

DARKON

Dark On (--> Funktion 99)

DELLINE

Delete Line (--> Funktion 28)

END

Ende des Programmes 'ALFA'.

ENDIF

Ende einer IF-Bedingung (--> IF3, IF4, IF5). Falls 'ENDIF' ohne ein vorhergehendes 'IF' angetroffen wird, erfolgt keine Aktion (d.h., die Zeile wird überlesen). IF-Bedingungen dürfen nicht verschachtelt werden.

FILE oder DISK

Alle weiteren Kommandos werden von der beim Aufruf des Programmes angegebenen Datei eingelesen. Falls beim Aufruf keine Datei angegeben wurde, so bleibt dieses Kommando ohne Wirkung.
Dieses Kommando hat nur dann einen Sinn, wenn von einer Datei mit dem Befehl 'CONSOLE' auf die Konsole umgeschaltet wurde.

FUNCTION n1 oder FUNKTION n1

Es wird die Funktion n1 auf der VIDEO I ausgelöst. Dabei sind nur solche Funktionen erlaubt, die keine weiteren Parameter benötigen.

GERMAN

Es wird der deutsche Zeichensatz selektiert (--> Funktion 54).

HALF

Das Attribut 'halbe Helligkeit' wird eingeschaltet
(--> Funktion 74).

IDENT oder IDENTS

Für alle auf dem Bildschirm darstellbaren Zeichen wird die Identität wieder hergestellt, d.h., es wird 256 mal die Funktion 55 mit n1=n2=0..255 ausgeführt. Außerdem werden alle Attribute zurückgesetzt (--> Funktion 72).

IDENTK

Für alle Tasten der Tastatur wird die Identität wieder hergestellt, d.h., es wird 256 mal die Funktion 56 mit n1=n2=0..255 ausgeführt.

IF3

Alle Kommandos zwischen 'IF3' und 'ENDIF' werden nur dann ausgeführt, wenn die VIDEO I mit einer Firmware Version 3 oder Version 5 ausgestattet ist.

IF4

Alle Kommandos zwischen 'IF4' und 'ENDIF' werden nur dann ausgeführt, wenn die VIDEO I mit einer Firmware Version 4 oder Version 5 ausgestattet ist.

IF5

Alle Kommandos zwischen 'IF5' und 'ENDIF' werden nur dann ausgeführt, wenn die VIDEO I mit einer Firmware Version 5 ausgestattet ist.

INSLINE

Insert Line (--> Funktion 27)

INVERS

Das Attribut 'inverse Zeichendarstellung' wird eingeschaltet (--> Funktion 80).

LD6845 n1 n2

Das Register n1 des 6845 wird mit dem Code n2 geladen (--> Funktion 94).

LDSTR1 n1 text

Die Funktionstaste mit dem Code n1 (256 > n1 > 127) wird mit dem String text belegt. Innerhalb des Textes text sind beliebig viele Leerzeichen erlaubt. Am Anfang und am Ende des Textes werden jedoch alle Leerzeichen entfernt. Der String wird **nicht** mit einem <CR> abgeschlossen.

LDSTR2 n1 text

Die Funktionstaste mit dem Code n1 (256 > n1 > 127) wird mit dem String text belegt. Innerhalb des Textes text sind beliebig viele Leerzeichen erlaubt. Am Anfang und am Ende des Textes werden jedoch alle Leerzeichen entfernt. Der String wird mit einem <CR> abgeschlossen.

LF

Line Feed (--> Funktion 4)

NEW n1 n2 n3 n4

In dem durch (n1..n4) beschriebenen Fenster werden alle Zeichen ausgelesen und erneut wieder eingeschrieben. Dies kann z.B. dazu benutzt werden, ein Fenster mit gleichem Inhalt, aber anderen Attributen darzustellen. Dazu müssen lediglich vorher die gewünschten neuen Attribute selektiert werden. (n1,n2) beschreibt die linke obere Ecke und (n3,n4) die rechte untere Ecke des Fensters jeweils in der Reihenfolge (Zeile, Spalte).

NORMAL

Alle Attribute werden gelöscht (--> Funktion 72).

PGACTUAL

Page Actual (--> Funktion 31)

PGDOWN

Page Down (--> Funktion 32)

PGUP

Page Up (--> Funktion 33)

RAHMEN1 n1 n2 n3 n4

'ALFA' zeichnet einen Rahmen (Rechteck) auf der VIDEO I. (n1,n2) beschreibt die linke obere Ecke und (n3,n4) die rechte untere Ecke (jeweils in der Reihenfolge: Zeile, Spalte). Der Rahmen wird mit einfacher Strichdicke gezeichnet.

RAHMEN2 n1 n2 n3 n4

'ALFA' zeichnet einen Rahmen (Rechteck) auf der VIDEO I. (n1,n2) beschreibt die linke obere Ecke und (n3,n4) die rechte untere Ecke (jeweils in der Reihenfolge: Zeile, Spalte). Der Rahmen wird mit Doppelstrichen gezeichnet, wenn 24 oder 25 Zeilen mit 80/96 oder 132 Zeichen pro Zeile aktiv sind. Andernfalls wird auch dieser Rahmen nur mit einfacher Strichdicke gezeichnet.

RAHMEN3 n1 n2 n3 n4

'ALFA' zeichnet einen Rahmen (Rechteck) auf der VIDEO I. (n1,n2) beschreibt die linke obere Ecke und (n3,n4) die rechte untere Ecke (jeweils in der Reihenfolge: Zeile, Spalte). Der Rahmen wird mit dem 'DEL'-Zeichen (Code = 7FH) gezeichnet.

RESET

VIDEO I wird neu initialisiert (--> Funktion 41).

SCRUP

Scroll Up (--> Funktion 30)

SCRDN

Scroll Down (--> Funktion 29)

TEXT text

Der mit diesem Kommando spezifizierte Text wird ab der augenblicklichen Cursor-Position an die VIDEO I ausgegeben. Innerhalb des Textes text sind beliebig viele Leerzeichen erlaubt. Am Anfang und am Ende des Textes werden jedoch alle Leerzeichen entfernt.

Falls in der ersten Spalte ein Leerzeichen steht, so wird das ebenfalls als TEXT-Kommando interpretiert, d.h. jede Zeile, die mit einem Leerzeichen beginnt, wird als TEXT-Kommando interpretiert.

Eine leere Zeile bewirkt eine Ausgabe von <CR> und <LF>. Falls in einer Zeile jedoch mindestens ein Zeichen enthalten ist, erfolgt keine Ausgabe von <CR> und <LF>.

UNDERLIN

Das Attribut 'unterstrichene Zeichendarstellung' bzw. 'zweiter Zeichensatz aktiv' wird eingeschaltet (--> Funktion 76).

US

Es wird der amerikanische Zeichensatz selektiert
(--> Funktion 53).

VIDIN

'ALFA' wartet auf eine Eingabe von der VIDEO I-Tastatur. Dabei ist es beliebig, welche Taste betätigt wird. Das eingegebene Zeichen wird nicht verwertet.

VERSION

'ALFA' gibt seine eigene Versionsnummer und die Versionsnummer der VIDEO I über die augenblickliche Konsole des Betriebssystems aus.

WAIT n1

'ALFA' führt bei einem Systemtakt von 4MHz eine Warteschleife von n1 Millisekunden aus. Bei anderen Taktfrequenzen ist die Wartezeit entsprechend höher oder kleiner.

WDCLEAR n1 n2 n3 n4

In dem durch (n1..n4) beschriebenen Bildschirmfenster wird der Text gelöscht (--> Funktion 97). (n1,n2) beschreibt die linke obere Ecke und (n3,n4) die rechte untere Ecke des Fensters in der Form (Zeile, Spalte).

WDSCRDN n1 n2 n3 n4

In dem durch (n1..n4) beschriebenen Bildschirmfenster wird der Text um eine Zeile nach unten verschoben (--> Funktion 95). (n1,n2) beschreibt die linke obere Ecke und (n3,n4) die rechte untere Ecke des Fensters in der Form (Zeile, Spalte).

WDSCRUP n1 n2 n3 n4

In dem durch (n1..n4) beschriebenen Bildschirmfenster wird der Text um eine Zeile nach oben verschoben (--> Funktion 96). (n1,n2) beschreibt die linke obere Ecke und (n3,n4) die rechte untere Ecke des Fensters in der Form (Zeile, Spalte).

WINDOW n1 n2 n3 n4

Der in den nachfolgenden (n3-n1+1) Zeilen folgende Text wird in das durch n1..n4 beschriebene Fenster eingeschrieben. (n1,n2) beschreibt die Zeile und Spalte der linken oberen Ecke und (n3,n4) die Zeile und Spalte der rechten unteren Ecke.

Alle in dem Fenster anzuzeigenden Textzeilen müssen mit einem Leerzeichen in der ersten Spalte beginnen (siehe auch unter 'TEXT'). Falls in der ersten Spalte ein neues Kommando beginnt, wird der 'WINDOW'-Befehl sofort abgebrochen.

Z12132

VIDEO I geht auf die Darstellung mit 12 Zeilen und 132 Zeichen pro Zeile (--> Funktion 106) über.

Z1240

VIDEO I geht auf die Darstellung mit 12 Zeilen und 40 Zeichen pro Zeile (--> Funktion 111) über.

Z1248

VIDEO I geht auf die Darstellung mit 12 Zeilen und 48 Zeichen pro Zeile (--> Funktion 110) über.

Z1266

VIDEO I geht auf die Darstellung mit 12 Zeilen und 66 Zeichen pro Zeile (--> Funktion 109) über.

Z1280

VIDEO I geht auf die Darstellung mit 12 Zeilen und 80 Zeichen pro Zeile (--> Funktion 108) über.

Z1296

VIDEO I geht auf die Darstellung mit 12 Zeilen und 96 Zeichen pro Zeile (--> Funktion 107) über.

Z24132

VIDEO I geht auf die Darstellung mit 24 Zeilen und 132 Zeichen pro Zeile (--> Funktion 50) über.

Z2440

VIDEO I geht auf die Darstellung mit 24 Zeilen und 40 Zeichen pro Zeile (--> Funktion 105) über.

Z2448

VIDEO I geht auf die Darstellung mit 24 Zeilen und 48 Zeichen pro Zeile (--> Funktion 104) über.

Z2466

VIDEO I geht auf die Darstellung mit 24 Zeilen und 66 Zeichen pro Zeile (--> Funktion 103) über.

Z2480

VIDEO I geht auf die Darstellung mit 24 Zeilen und 80 Zeichen pro Zeile (--> Funktion 52) über.

Z2496

VIDEO I geht auf die Darstellung mit 24 Zeilen und 96 Zeichen pro Zeile (--> Funktion 51) über.

Z25132

VIDEO I geht auf die Darstellung mit 25 Zeilen und 132 Zeichen pro Zeile (--> Funktion 100) über.

Z2580

VIDEO I geht auf die Darstellung mit 25 Zeilen und 80 Zeichen pro Zeile (--> Funktion 102) über.

Z2596

VIDEO I geht auf die Darstellung mit 25 Zeilen und 96 Zeichen pro Zeile (--> Funktion 101) über.

10. Der Grafikeditor 'GRED'

'GRED' ist ein Programm, das dem Anwender zeigen soll, wie die Funktionen der VIDEO I im Grafik-Modus zu nutzen sind. Es stellt zudem ein Hilfsmittel dar, um Grafiken in der Größe eines Rasterfeldes von 512 (256) mal 768 Punkten interaktiv zu erstellen.

'GRED' ist auf der Diskette mit den VIDEO I-Hilfsprogrammen als Quelltext enthalten ('GRED.PAS' und 'GREDIO.ASM'). Eine Modifikation zur eigenen Benutzung ist gestattet und erwünscht.

10.1 Das Programm 'GRED'

'GRED' ist in der Programmiersprache Pascal geschrieben. Der Quelltext weicht an einigen Stellen vom Standard-Pascal ab und richtet sich nach der Pascal-Definition zum Compiler PROPAS der Firma Prospero Software England.

Das Hauptprogramm besteht aus einem Initialisierungsteil und einem Kommandointerpreter. Die einzelnen Prozeduren und Funktionen führen die Kommandos des Anwenders aus.

Die Schnittstelle zum Anwender bildet die Funktion GETC. Sie holt die Kommandos von der Tastatur. Dabei wird unterschieden zwischen der Tastatur, die an die VIDEO I angeschlossen ist, und der Konsole. Standardmäßig wird die Tastatur der VIDEO I abgefragt. Soll stattdessen mit der Konsole als Eingabemedium gearbeitet werden, so muß beim Programmstart der Parameter 'KBD:' angegeben werden.

Alle Ausgaben (z.B. Abfrage nach der Eingabedatei) gelangen auf die VIDEO I. Die Konsole dient nicht als Ausgabemedium.

Für den Fall, daß die VIDEO I als Konsole betrieben wird, erfolgen alle Eingaben über die Tastatur der VIDEO I.

Wird statt 'KBD:' der Name einer Datei als Parameter beim Programmaufruf angegeben, so holt 'GRED' alle Eingaben von der Datei. Am Dateiende schaltet 'GRED' auf die VIDEO-I-Tastatur als Eingabemedium um.

10.1.1 Schnittstelle zur VIDEO I

Der Zugriff zur VIDEO I erfolgt über die Prozedur VIDOUT und der Funktion VIDIN. Beides sind Unterprogramme, die in der Z80-Assembler-Sprache geschrieben sind. Die Parameterübergabe erfolgt nach der vom Compiler vorgegebenen Konvention. Sie weicht deshalb von den Beispielen ab, die an anderer Stelle dieses Handbuches für die Schnittstelle zu einem FORTRAN-80-Programm aufgeführt sind.

Die Prozedur VIDOUT übergibt ein Byte an die VIDEO I.

Die Funktion VIDIN übergibt ein Byte von der VIDEO I.

Die Unterprogramme sprechen die VIDEO I unter der Port-Adresse an, die für den Auslieferzustand definiert ist:

0B8H	für das Datenport
0B9H	für das Statusport.

10.1.2 Schnittstelle zum Drucker

Der Drucker wird über einen BIOS-Aufruf im Unterprogramm LISTER angesprochen. Die Prozedur übergibt ein Byte vom aufrufenden Programm an den Drucker.

Als Drucker ist ein Seikosha Business Printer BP-5420A vorgesehen worden. Bei Benutzung eines anderen Druckers müssen die Grafik-Steuerzeichen des Druckers im Programmteil 'HARDCO' neu angepaßt werden.

10.1.3 Abspeicherformat auf der Magnetplatte

Die Bildpunkte werden byteweise aus der VIDEO I ausgelesen und mit zwei Byte abgespeichert, wenn mindestens ein Bit in dem Byte von Null verschieden ist. Das erste Byte enthält immer eine Null, während das zweite Byte mit dem Byte von der VIDEO I identisch ist. Aufeinanderfolgende Byte, die nur Nullen enthalten, werden mit ihrer Anzahl in einem Byte abgespeichert. Bis zu 255 Byte mit Nullen werden so zu einem Byte komprimiert.

Das Kodierverfahren führt dazu, daß die Länge der Datei, auf die Bilddaten geschrieben werden, abhängig ist vom Bildinhalt. Ein völlig leeres Bild wird in einem CP/M-Sektor (128 Byte) abgespeichert. Ein Bild, in dem jeder achte Bildpunkt gesetzt ist, beansprucht bei einem Bildformat von 256 mal 768 Punkten 52KByte auf der Magnetplatte.

Das Kodierverfahren wurde gewählt, da Bildinhalte mit vielen leeren Bildteilbereichen häufiger vorkommen als solche, in denen fast jeder Bildpunkt gesetzt ist. Es ergibt sich in den meisten Fällen eine geringere Dateilänge, als bei der unkodierten Abspeicherung.

10.1.4 Anmerkung

Das Programm 'GRED' ist ein Beispielprogramm. Eine Gewähr für die Richtigkeit, Vollständigkeit und Perfektion des Programmes oder der Beschreibung wird nicht geleistet. Unvollständige oder unbefriedigende Funktionen mag der Anwender selbst ergänzen oder verbessern.

10.2 Aufruf des Programmes 'GRED'

'GRED' kann mit einer der beiden folgenden Formen aufgerufen werden:

GRED
GRED name

Im ersteren Fall wird für Eingaben an 'GRED' die VIDEO I-Tastatur unter Umgehung des Betriebssystems benutzt.

Im zweiten Fall werden alle Kommandos aus der Datei name ausgelesen. Wenn die Menge der Kommandos aus der Datei name erschöpft ist, erwartet 'GRED' weitere Kommandos von der VIDEO I-Tastatur. Für name kann auch die fest eingestellte Bezeichnung 'KBD:' gewählt werden. In diesem Fall werden alle Eingaben von der augenblicklichen Konsole des Betriebssystems erwartet.

Ausgaben erfolgen auf jeden Fall unter Umgehung des Betriebssystems direkt auf die VIDEO I.

Falls auf dem augenblicklichen Laufwerk (current drive) des Betriebssystems eine Datei mit dem fest eingestellten Namen 'CSET' (ohne Typangabe) gefunden werden kann, wird diese als Zeichensatz interpretiert und in die VIDEO I geladen.

'CSET' muß eine von dem Programm 'TRCHAR' erzeugte Datei sein und dementsprechend zwei Zeichensätze enthalten. Beide möglichen ladbaren Zeichensatzpaare der VIDEO I werden dann mit 'CSET' vorbesetzt.

Nach dem Aufruf von 'GRED' muß als erstes der Grafik-Modus selektiert werden. Dazu sind folgende 4 Kommandos möglich:

6 Grafik-Modus 1 wird selektiert, aber auf der VIDEO I nicht initialisiert. Eine sich bereits auf der VIDEO I befindliche Grafik kann weiter editiert werden.

7 Grafik-Modus 2 wird selektiert, aber auf der VIDEO I nicht initialisiert. Eine sich bereits auf der VIDEO I befindliche Grafik kann weiter editiert werden.

8 Grafik-Modus 1 wird selektiert und auf der VIDEO I initialisiert. Der Bildschirm wird dabei gelöscht.

9 Grafik-Modus 2 wird selektiert und auf der VIDEO I initialisiert. Der Bildschirm wird dabei gelöscht.

Wenn sich die VIDEO I noch im ALFA-Modus befindet, ist an dieser Stelle die Eingabe von 8 oder 9 erforderlich. Befindet sich die VIDEO I bereits im Grafik-Modus 1 ist außer 8 und 9 auch die Eingabe von 6 möglich. Wenn sich die VIDEO I dagegen bereits im Grafik-Modus 2 befindet, ist außer 8 und 9 an dieser Stelle nur die Eingabe von 7 erlaubt.

WARNUNG: Eine Eingabe von 6 oder 7 ist nur dann erlaubt, wenn sich die VIDEO I tatsächlich im Grafik-Modus 1 (nur Eingabe von 6 erlaubt) bzw. im Grafik-Modus 2 (nur Eingabe von 7 erlaubt) befindet! In allen anderen Fällen arbeitet 'GRED' fehlerhaft bis hin zum Systemabsturz!

10.3 Die Kommandos des 'GRED'

10.3.1 Kommandos, die den Cursor bewegen

Der Cursor läßt sich punktweise nach rechts, links, oben und unten bewegen. Zur Überbrückung größerer Entfernungen kann der Cursor auch in Schritten von 10 Bildpunkten bewegt werden. Bei der Bewegung des Cursors, verändert sich der Bildinhalt auf dem zurückgelegten Weg entsprechend der Einstellung für die Bildpunktmanipulation.

E	bewegt den Cursor um einen Bildpunkt nach oben
X	bewegt den Cursor um einen Bildpunkt nach unten
D	bewegt den Cursor um einen Bildpunkt nach rechts
S	bewegt den Cursor um einen Bildpunkt nach links
R	bewegt den Cursor um zehn Bildpunkte nach oben
C	bewegt den Cursor um zehn Bildpunkte nach unten
F	bewegt den Cursor um zehn Bildpunkte nach rechts
A	bewegt den Cursor um zehn Bildpunkte nach links

Der Cursor läßt sich schnell auch über größere Strecken bewegen, jedoch ohne daß der Bildinhalt auf der zurückgelegten Strecke verändert wird.

^Q E	setzt den Cursor an den oberen Bildrand
^Q X	setzt den Cursor an den unteren Bildrand
^Q D	setzt den Cursor an den rechten Bildrand
^Q S	setzt den Cursor an den linken Bildrand
^Q B	setzt den Cursor an die Marke 'B'
^Q K	setzt den Cursor an die Marke 'K'
^Q 1	setzt den Cursor an die Marke '1'
^Q 2	setzt den Cursor an die Marke '2'
^Q 3	setzt den Cursor an die Marke '3'

Bevor der Cursor jedoch an eine Marke positioniert werden kann, muß die Marke erst definiert werden:

^K B	definiert an der momentanen Position die Marke 'B'
^K K	definiert an der momentanen Position die Marke 'K'
^K 1	definiert an der momentanen Position die Marke '1'
^K 2	definiert an der momentanen Position die Marke '2'
^K 3	definiert an der momentanen Position die Marke '3'

10.3.2 Bildinhalt abspeichern und laden

Der gesamte Bildinhalt läßt sich auf eine Datei schreiben. Dazu muß das Kommando:

^K W

eingegeben werden. 'GRED' fragt dann nach dem Namen der Zieldatei und beginnt mit der Abspeicherung. Eine bereits existierende Datei mit dem eingegebenen Namen wird überschrieben.

Ein einmal abgespeicherter Bildinhalt kann wieder geladen werden, wenn das Kommando:

^K R

eingegeben wird. 'GRED' fragt wieder nach einem Dateinamen. Falls keine Datei mit dem eingegebenen Namen existiert, fragt 'GRED' nach einem anderen Namen und löscht dazu den alten Namen im Anzeigefeld.

Das Abspeichern und das Laden kann eine längere Zeit in Anspruch nehmen. Während des Abspeicherns verschwindet der Cursor vom Bildfeld. Wenn der Cursor wieder erscheint ist die Abspeicherung beendet.

10.3.3 Zeichensatz laden und auswählen

Beim Aufruf des Programmes 'GRED' kann automatisch bereits ein Zeichensatz geladen werden (siehe 10.2).

Während des Betriebes von 'GRED' kann mit dem folgenden Kommando jederzeit erneut ein Zeichensatz geladen werden:

^Lname<CR>

name ist der Dateiname des Zeichensatzes, der hier geladen werden soll. Für den Aufbau der Datei und das Laden gilt hier das gleiche, wie unter 10.2 gesagte.

Mit dem Befehl

^Zn

mit n = 'C', 'c', 'D', 'd', 'F' oder 'f'

wird der zugeordnete Zeichensatz (siehe 3.3.4.2) angewählt und nachfolgend für alle Textdarstellungen im 'GRED' benutzt.

10.3.4 Linien zeichnen

Linien lassen sich mit den Kommandos zur Cursorbewegung ziehen. Es ist jedoch auch möglich, mit einem Kommando eine Verbindungslinie von der augenblicklichen Cursorposition zu einer Marke ('1', '2'

oder '3') zu ziehen. Die Linie wird entsprechend der Einstellung für die Bildpunktmanipulation gezeichnet.

Kommando: V 1, V 2 oder V 3

Mit diesem Kommando lassen sich auch Linien mit beliebiger Schräglage zeichnen.

Nach Ausführung des Kommandos: O

wird bei jeder Cursor-Bewegung eine Linie von der augenblicklichen Cursorposition zu der Marke '1' gezogen. Auch hier wird eine Linie entsprechend der Einstellung für die Bildpunktmanipulation gezeichnet.

Dieses Kommando läßt sich wieder abschalten mit dem Kommando: P

10.3.5 Bildinhalt auf dem Drucker darstellen

Der gesamte Bildinhalt kann mit einem Befehl an den Drucker übermittelt werden. Der Drucker muß über das Betriebssystem als Lister erreichbar sein.

Kommando: ^K P

Die Steuerzeichen für die grafische Darstellung auf dem Drucker können im Programmteil HARDCO für das jeweilige Gerät verändert werden.

10.3.6 Text eingeben

Nach dem Kommando

^T

werden alle eingegebenen Zeichen als Textzeichen auf dem Bildschirm dargestellt bis entweder ein ^Ö oder ein ^Ü eingegeben wird.

Die Position des Grafik-Cursors ändert sich dabei nicht.

10.3.7 Voreinstellung für die Bildpunktmanipulation

Der Bildinhalt läßt sich punktweise verändern (z. B. durch Cursorbewegungen). In welcher Art dies geschieht, hängt von der gewählten Manipulationsart ab.

Das Kommando

+

bewirkt, daß nachfolgend alle Punkte gesetzt werden.

Nach dem Kommando

*

werden alle Punkte invertiert und nach dem Kommando

-

werden Bildpunkte gelöscht.

Damit der Cursor bewegt werden kann, ohne daß der Bildinhalt verändert wird, kann mit dem Kommando

!

die Manipulation unterdrückt werden.

10.3.8 Grafik Modus wechseln

Grafik Modus 1 (768 x 512) einstellen: 8
Grafik Modus 2 (768 x 256) einstellen: 9

Beim Wechseln in den jeweils anderen Grafik-Modus wird der Bildschirm gelöscht und neu initialisiert.

10.3.9 'GRED' beenden

Es gibt 3 Möglichkeiten 'GRED' zu beenden:

^K D

'GRED' speichert die bearbeitete Grafik analog zum Befehl '^KW' auf der Magnetplatte ab und beendet die Bearbeitung der augenblicklichen Grafik. Der Grafik-Modus der VIDEO I bleibt eingeschaltet.

^K Q

'GRED' bricht die Bearbeitung der augenblicklichen Grafik ab, ohne daß das Bild auf die Magnetplatte zurückgeschrieben wird. Der Grafik-Modus der VIDEO I bleibt eingeschaltet.

0 (Null)

'GRED' bricht die Bearbeitung der augenblicklichen Grafik ab und schaltet in den ALFA-Modus der VIDEO I zurück.

Falls 'GRED' mit '^KD' oder 'KQ' beendet wurde, kann die Bearbeitung der gleichen Grafik mit 'GRED' wieder aufgenommen werden (siehe Aufruf von 'GRED').

11. Disketteninhalt der Diskette 'Hilfsprogramme zur VIDEO I

Auf der Diskette 'Hilfsprogramme zur VIDEO I' sind folgende Dateien enthalten:

ALFA.COM				--> Kap. 9
GRED.COM	**GRED.PAS**	**GREDIO.ASM**		--> Kap. 10
TERM5.ASM				--> Kap. 5
TRCHAR.COM	**LDCHAR.COM**			--> Kap. 8
CHAR.COM	**CHRBREIT.COM**	**CHRHOCH.COM**	**CHRHB.COM**	--> Kap. 7
CHRDEL.COM	**CHRCOMP.COM**	**Z96.DAT**	**Z132.DAT**	--> Kap. 7
Z48.DAT	**Z66.DAT**	**Z1296.DAT**	**Z12132.DAT**	--> Kap. 7
Z1248.DAT	**Z1266.DAT**	**ZTAB48.DAT**	**ZTAB66.DAT**	--> Kap. 7
DEL48.DAT				--> Kap. 7

HCOPY1.ASM

HCOPY1 ist ein Programm zur Erstellung einer Grafik-Hardcopy auf einem ITOH 8510A. Das Programm benutzt die Hardcopy-Funktion der VIDEO I. Es kann mit dem 'RMAC'-Assembler von Digital-Research (gehört zum CP/M PLUS) assembliert werden. Weitere Einzelheiten können dem ausführlich kommentierten Quelltext entnommen werden.

HCOPY2.ASM

HCOPY2 ist ein Programm zur Erstellung einer Grafik-Hardcopy auf einem ITOH 8510A oder auf einem EPSON FX80. Das Programm benutzt die Funktion 'Byteauslesen' der VIDEO I und erstellt eine um 90° gedrehte Hardcopy auf dem Drucker. Auf dem ITOH 8510A und auf dem Epson FX80 sind nur bei um 90° gedrehten Hardcopys korrekte Seitenverhältnisse auf dem Drucker möglich. Das Programm kann mit dem 'RMAC'-Assembler von Digital-Research (gehört zum CP/M PLUS) assembliert werden. Weitere Einzelheiten können dem ausführlich kommentierten Quelltext entnommen werden.

CHRSET.COM

CHRSET ist ein Programm zur Anzeige eines augenblicklichen Zeichensatzes auf der VIDEO I.

L6845.COM

Mit dem Programm L6845 können verschiedene Werte für die Register des 6845 interaktiv verschoben werden. Gleichzeitig kann die Auswirkung sofort auf dem Bildschirm beobachtet werden.
Mit Hilfe dieses Programmes ist es für den erfahrenen Programmierer leicht möglich, die optimale Initialisierung für einen bestimmten Monitor herauszufinden. Diese Werte können dann über die Funktion 94 der VIDEO I beim 'Hochfahren' des Betriebssystems in die Register des 6845 eingeschrieben werden.
Alle augenblicklichen Werte der 6845-Register werden auf dem Bildschirm angezeigt und können durch Betätigen der ebenfalls angezeigten Tasten incrementiert oder decrementiert werden.
Dieses Programm setzt Kenntnisse über den VIDEO-Controller 6845 von Motorola voraus!

***.DEM**

Diverse Demos als Eingabedatei für das Programm 'ALFA'.

***.GRF**

Diverse Demos für den Grafik-Editor 'GRED'. Diese Demos können im 'GRED' mit dem Befehl '^KR' eingelesen werden.

12. Glossar

Austastlücke

siehe BAS-Signal

BAS-Signal

BAS steht für **B**ild-**A**ustast-**S**ynchron-Signal
Unter Bildsignal wird das Signal zur Erzeugung des sichtbaren Bildes auf einem Monitor verstanden.
Jeweils nach Ende einer Bildschirmzeile muß der Elektronenstrahl der Bildröhre an den Anfang der nächsten Zeile zurückgeführt werden. Nach Ende eines Elektronenstrahl-Bildes muß der Strahl an den Anfang des Bildes zurückgeführt werden. Während dieser Strahlrückführungszeiten wird der Elektronenstrahl dunkel gesteuert oder anders ausgedrückt: 'ausgetastet'. Das dafür notwendige Austastsignal muß von einem Video-Controller zur Verfügung gestellt werden. Während dieser 'Austastlücke' im Bildsignal greift der Video-Controller nicht auf den Bildspeicher zu, so daß der Zugriff durch eine CPU auf der VIDEO-Platine möglich wird.
Die Strahlablenkung des Monitors muß bezüglich Zeilenanfang und Bildanfang mit den Werten des jeweiligen Bildsignales übereinstimmen. Zu diesem Zweck werden vom Video-Controller Synchron-Impulse erzeugt.
Das Bildsignal, das Austastsignal und das Synchronsignal werden auf der VIDEO I zum BAS-Signal gemischt.

c.p.

Abkürzung für 'Cursorposition'

Cursorposition

Cursorposition ist die durch x- und y-Koordinaten beschreibbare Position auf dem Bildschirm, an der die nächste Bildschirm-Manipulation stattfinden würde, wenn zwischendurch keine neue Cursorposition angewählt wird. Mit x werden dabei horizontale Positionen und mit y vertikale Positionen bezeichnet. Der Nullpunkt befindet sich links oben auf dem Bildschirm.
Im ALFA-Modus wird mit der Cursorposition eine vollständige Zeichenposition auf dem Bildschirm beschrieben. Der mögliche Wertebereich von x ist 0..131 und von y 0..24.
Im Grafik-Modus wird mit der Cursorposition ein einzelner Punkt auf dem Bildschirm beschrieben. Der mögliche Wertebereich von x beträgt hier 0..767 und von y 0..511.

interlace

siehe Zeilensprungverfahren

Kenncode

Der Code, der an die VIDEO I übertragen werden muß, um eine bestimmte Funktion auszulösen. Der Kenncode ist abhängig von der auszulösenden Funktion und von der jeweiligen Terminal-Emulation.

Beispiel: Der Kenncode für die Funktion 'Cursor Home' ist bei der Terminal-Emulation 1 alternativ das Zeichen <SOH> oder die ESC-Sequenz <ESC> 'X' 36.
Bei der Terminal-Emulation 2 besteht der Kenncode für diese Funktion alternativ aus der ESC-Sequenz <ESC> <DC2> oder der ESC-Sequenz <ESC> 'X' 36.
Bei der Terminal-Emulation 3 besteht der Kenncode für diese Funktion nur aus dem Zeichen <FS>.

LSB

'least significant bit'
Das niedrigwertigste Bit in einem Byte oder 16-Bit-Wort (also Bit 0, wenn die Zählung mit 0 beginnt).

MSB

'most significant bit'
Das höchstwertigste Bit in einem Byte oder 16-Bit-Wort (also Bit 7 bzw. Bit 15, wenn die Zählung mit Bit 0 beginnt).

non interlace

Gegenteil von interlace

Vektor

In diesem Handbuch verstehen wir ohne Anspruch auf mathematische Exaktheit unter einem Vektor eine durch Anfangs- und Endpunkt beschriebene Teilmenge einer Geraden im zweidimensionalen Raum. Im Grafik-Modus wird ein Vektor also durch eine Verbindungslinie seines Anfangs- und Endpunktes auf dem Bildschirm dargestellt.

Zeilensprung

siehe Zeilensprungverfahren

Zeilensprungverfahren

Die Fernsehnorm sieht 625 Bildschirmzeilen pro Bild und 25 Bilder pro Sekunde vor. 25 Bilder pro Sekunde werden jedoch als flimmerndes Bild empfunden. Daher werden bei einem Fernsehbild nicht 25 (Voll-)Bilder zu 625 Zeilen übertragen, sondern 50 Halb-Bilder zu 312,5 Zeilen pro Halbbild. Durch die halben Zeilen am Ende bzw. Anfang eines Halbbildes wird bei der Darstellung auf einem Bildschirm erreicht, daß das jeweils nachfolgende Bild um eine Zeile versetzt auf dem Bildschirm erscheint. Beide Halb-Bilder sind somit kammartig ineinander verzahnt. Bei üblichen großflächigen bewegten Fernsehbildern wird dadurch ein weitgehend flimmerfreies Bild erzeugt. Lediglich bei den im Fernsehbild seltenen waagerechten Grenzen zwischen weiß und schwarz ist ein Flimmereffekt zu bemerken.
In der Datentechnik folgt aber sehr oft nach einer 'weißen' Zeile eine 'schwarze', so daß dieser Flimmereffekt des Zeilensprungverfahrens sehr störend wirken kann. Durch eine Änderung der Ansteuersignale kann jedoch erreicht werden, daß beide 'Halb'-Bilder nicht mehr um eine Zeile versetzt auf dem Bildschirm erscheinen, sondern deckungsgleich dargestellt werden. Wenn nun gleichzeitig in beiden 'Halb'-Bildern die gleiche Bildinformation enthalten ist, werden diese 'Halb'-Bilder zu 'Voll'-Bildern mit einer Bildfrequenz von 50 Bildern pro Sekunde. Bei dieser Bildfrequenz kann nun kein Flimmern mehr beobachtet werden. Allerdings muß man jetzt beachten, daß ohne Zeilensprung nur jede 2. Bildschirmzeile beschrieben und dadurch die Auflösung halbiert wird. Der Abstand zwischen 2 aufeinanderfolgenden Zeilen ist bei der Darstellung ohne Zeilensprungverfahren etwa doppelt so groß wie mit Zeilensprungverfahren.

Anhang A:

Kurzübersicht der Steuersequenzen im Alfa-Modus (mit Firmware 1.1xx)

Alle Funktionen können über ESC 'X' und anschließender Funktionsnummer (3 bis 115 binär) im Terminal 1 und 2 erreicht werden. Darüber hinaus sind einige Funktionen auch noch über andere Steuersequenzen erreichbar. Terminal 1 verhält sich weitgehend kompatibel zu einem ADDS Viewpoint (im WordStar-Menue enthalten). Terminal 2 verhält sich weitgehend kompatibel zu einem Geveke VISA 30/40 (Hazeltine 1510 mit ESC statt Tilde = '~' als "leadin code"). Terminal 3 verhält sich weitgehend kompatibel zu einem ADM 3a mit Erweiterungen (ähnlich info-S VIDEO 7). Terminal 4 ist nicht belegt (die Tabellen im EPROM sind mit 0FFH belegt und können nachprogrammiert werden). Die Tabelle für Terminal 5 muß zuerst vom Computersystem geladen werden.

Keyboard-Funktionen können nur ausgelöst werden, wenn auf dem Keyboard Tasten mit gesetztem 8. Bit zur Verfügung stehen. In der Keyboard-Spalte sind die Tastencodes der Tasten angegeben, die nacheinander betätigt werden müssen, um die gewünschte Funktion auszulösen. Falls keine Tasten mit gesetztem 8. Bit zur Verfügung stehen, muß zur Ausnutzung der Keyboard-Funktionen erst eine Tastaturumkodierung mit der Funktion 56 vorgenommen werden.

Funktion auf jk82 VIDEO I	Nr.	Terminal 1	Terminal 2	Terminal 3	Keyboard
RESET CURSOR STACK	115	--	--	--	--
SWAP CURSOR	114	--	--	--	--
POP CURSOR	113	--	--	ESC 'j'	--
PUSH CURSOR	112	--	--	ESC 'i'	--
40 CHARACTERS PER LINE (12 LINES)	111	--	--	ESC 'U'	--
48 CHARACTERS PER LINE (12 LINES)	110	--	--	--	--
66 CHARACTERS PER LINE (12 LINES)	109	--	--	--	--
80 CHARACTERS PER LINE (12 LINES)	108	--	--	--	--
96 CHARACTERS PER LINE (12 LINES)	107	--	--	--	--
132 CHARACTERS PER LINE (12 LINES)	106	--	--	--	--
40 CHARACTERS PER LINE (24 LINES)	105	--	--	ESC 'T'	--
48 CHARACTERS PER LINE (24 LINES)	104	--	--	--	--
66 CHARACTERS PER LINE (24 LINES)	103	--	--	--	--
80 CHARACTERS PER LINE (25 LINES)	102	--	--	--	80H, 80H, 89H
96 CHARACTERS PER LINE (25 LINES)	101	--	--	--	80H, 80H, 8AH
132 CHARACTERS PER LINE (25 LINES)	100	--	--	--	80H, 80H, 8BH
SCREEN DARK ON	99	--	--	--	--
SCREEN DARK OFF	98	--	--	--	--
CLEAR WINDOW	97	--	--	--	--
WINDOW SCROLL UP	96	--	--	--	--
WINDOW SCROLL DOWN	95	--	--	--	--
LOAD 6845 DIRECT	94	--	--	--	--
IGNORE NEXT CHAR	93	ESC '0'	--	--	--
DISPLAY TEST PATTERN	92	--	ESC '*'	--	80H, 80H, 84H
CLEAR KEYBOARD BUFFER	91	--	--	--	84H
KEYBOARD BUFFER ACTIVE	90	--	--	--	80H, 86H
KEYBOARD BUFFER NOT ACTIV	89	--	--	--	80H, 80H, 86H
RING BELL	88	BEL	BEL	BEL	80H
CURSOR ON	87	CAN	--	ESC '0'	80H, 80H, 88H
CURSOR BLINK ON	86	ESC 'c'	--	ESC 'M'	80H, 88H
CURSOR OFF	85	ETB	--	ESC 'P'	--

Funktion auf jk82 VIDEO I	Nr.	Terminal 1	Terminal 2	Terminal 3	Keyboard
KEYBOARD UNLOCK	84	ESC '6'	ESC ACK	ESC 'g'	wie Terminal
KEYBOARD LOCK	83	ESC '5'	ESC NAK	ESC 'y'	--
INVERS SCREEN ON	82	--	--	ESC 'I'	80H, 80H, 87H
INVERS SCREEN OFF	81	--	--	ESC 'H'	80H, 87H
CHAR INVERS ON	80	--	--	ESC '!' '4'	--
CHAR INVERS OFF	79	--	--	ESC '?' '4'	--
BLINKING CHAR ON	78	--	--	ESC '!' '3'	--
BLINKING CHAR OFF	77	--	--	ESC '?' '3'	--
UNDERLINE ON	76	--	--	ESC '!' '5'	--
UNDERLINE OFF	75	--	--	ESC '?' '5'	--
HALF INTENSITY ON	74	--	--	ESC '!' '2'	--
HALF INTENSITY OFF	73	--	--	ESC '?' '2'	--
NORMAL VIDEO	72	SI	ESC US	--	--
CHAR INVERS ONLY	71	--	--	--	--
BLINK ONLY	70	--	--	--	--
UNDERLINE ONLY	69	--	--	--	--
HALF INTENSITY ONLY	68	--	ESC EM	--	--
HALF INTENSITY BLINK	67	--	--	--	--
CHAR INVERS + HALF INTENSITY	66	SO	--	--	--
CHAR INVERS + BLINK	65	--	--	--	--
CHAR INVERS + HALF INT. + BLINK	64	--	--	--	--
UNDERLINE + HALF INTENSITY	63	--	--	--	--
UNDERLINE BLINK	62	--	--	--	--
UNDERLINE + HALF INT. + BLINK	61	--	--	--	--
IDENTIFICATION	60	ESC 0FFH	--	--	--
LOAD STRING	59	ESC 'a'	ESC 'a'	--	--
CLEAR STRING BUFFER	58	ESC 'b'	ESC 'b'	ESC 'p'	--
FORMFEED	57	FF	--	--	--
CHANGE CHARACTER CODE SET (KEYBOARD)	56	--	--	--	--
CHANGE CHARACTER CODE SET (SCREEN)	55	--	--	--	--
GERMAN CHARACTER SET	54	--	--	ESC 'Y'	80H, 85H
US CHARACTER SET	53	--	--	ESC 'X'	80H, 80H, 85H
80 CHARACTERS PER LINE (24 LINES)	52	ESC '8'	ESC '8'	ESC 'S'	80H, 80H, 81H
96 CHARACTERS PER LINE (24 LINES)	51	ESC '9'	ESC '9'	--	80H, 80H, 82H
132 CHARACTERS PER LINE (24 LINES)	50	ESC '7'	ESC '7'	--	80H, 80H, 83H
ENABLE TERMINAL 1	49	ESC 'q'	ESC 'q'	ESC '?' 49	80H, 81H
ENABLE TERMINAL 2	48	ESC 'r'	ESC 'r'	ESC '?' 48	80H, 82H
ENABLE TERMINAL 3	47	ESC 's'	ESC 's'	ESC '?' 47	80H, 83H
ENABLE TERMINAL 4	46	ESC 't'	ESC 't'	ESC '?' 46	80H, 84H
ENABLE TERMINAL 5	45	ESC 'u'	ESC 'u'	ESC '?' 45	--
LOAD TERMINAL 5	44	ESC 'v'	ESC 'v'	ESC '?' 44	--

Funktion auf jk82 VIDEO I	Nr.	Terminal 1	Terminal 2	Terminal 3	Keyboard
GRAFIC MODE 1	43	--	--	--	--
GRAFIC MODE 2	42	--	--	--	--
RESET VIDEO I	41	--	--	--	80H, 80H, 8CH
ERASE TO END OF SCREEN	40	ESC 'k'	ESC CAN	CAN	--
ERASE TO START OF SCREEN	39	ESC 'L'	ESC 'L'	--	--
ERASE SCREEN	38	DC2	--	--	--
ERASE SCREEN + CURSOR HOME	37	DC3	ESC FS	FF	--
CURSOR HOME	36	SOH	ESC DC2	FS	--
ERASE TO END OF LINE	35	ESC 'K'	ESC SI	ETB	--
ERASE LINE	34	--	--	RS	--
PAGE UP	33	ESC 'E'	--	--	81H
PAGE DOWN	32	ESC 'F'	--	--	82H
PAGE ACTUAL	31	ESC 'G'	--	--	83H
SCROLL UP	30	--	--	--	85H
SCROLL DOWN	29	--	--	--	86H
PARTIAL SCROLL UP	28	--	ESC DC3	SUB	--
PARTIAL SCROLL DOWN	27	--	ESC SUB	EM	--
SEND CURSOR ADDRESS					
WITHOUT OFFSET ==> X ; Y	26	--	ESC ENQ	--	--
SEND CURSOR ADDRESS					
WITH OFFSET OF 20H ==> X ; Y	25	--	--	--	--
SEND CURSOR ADDRESS					
WITHOUT OFFSET ==> Y ; X	24	--	--	--	--
SEND CURSOR ADDRESS					
WITH OFFSET OF 20H ==> Y ; X	23	--	--	ESC 'C'	--
SEND CHAR AT CURSOR	22	--	ESC '!'	ESC 'a'	--
CURSOR FORWARD	21	ACK	--	HT	--
CURSOR RIGHT	20	--	DLE	--	--
CURSOR LEFT	19	NAK	--	--	--
CURSOR UP	18	SUB	ESC FF	VT	--
CURSOR DOWN	17	--	ESC VT	--	--
RESERVED: DO NOT USE	16	--	--	--	--
RESERVED: DO NOT USE	15	--	--	--	--
VERTICAL CURSOR ADDRESSING	14	VT	--	--	--
HORIZONTAL CURSOR ADDRESSING	13	DLE	--	--	--
ABSOLUTE CURSOR ADDRESSING					
WITHOUT OFFSET ==> X ; Y	12	--	ESC DC1	--	--
ABSOLUTE CURSOR ADDRESSING					
WITH OFFSET 20H ==> X ; Y	11	--	--	--	--
ABSOLUTE CURSOR ADDRESSING					
WITHOUT OFFSET ==> Y ; X	10	--	--	--	--
ABSOLUTE CURSOR ADDRESSING					
WITH OFFSET 20H ==> Y ; X	9	ESC 'Y'	--	ESC '='	--
CARRIAGE RETURN + ERASE END OF LINE	8	--	--	--	--
CARRIAGE RETURN + LINE FEED	7	--	--	--	--
CARRIAGE RETURN	6	CR	CR	CR	--
REVERSE LINEFEED	5	--	--	--	--
LINEFEED	4	LF	LF	LF	--
BACKSPACE	3	BS	BS	BS	--

Anhang: B

Kurzübersicht der Befehle im Grafik-Modus (mit Firmware 1.1xx)

In beiden Grafik-Modi sind die Steuerbefehle der VIDEO I gleich. Der Wertebereich der X-Achse reicht von 0 bis 767 und der Y-Achse von 0 bis 511. Der Nullpunkt befindet sich in der linken oberen Ecke des Bildschirmes.

punktsequenz = ESC, xlow, xhigh, ylow, yhigh

nicht im Text-Modus:

Punktmanipulation	setzen	'+'
Punktmanipulation	löschen	'-'
Punktmanipulation	invertieren	'*'
Punkt	ändern	punktsequenz
Koordinate	setzen	'K', punktsequenz
Punkt	holen	'G', punktsequenz
Vektor	zeichnen	'V', punktsequenz1, punktsequenz2
Vektor von c.p.	zeichnen	'v', punktsequenz
Byte	setzen	'S'
Byte	holen	'R'
Hardcopy		'H', punktsequenz
Text-Modus	einschalten	'C','c','D','d','F' oder 'f'
Bildschirm	löschen	^J
Grafik-Modus	verlassen	'E'
Bildschirm dunkel		^Q
Bildschirm hell		^P

Text- und normaler Grafik-Modus:

Bildschirm	löschen	^L
Bildschirm löschen und Grafik-Modus einschalten		^Y
Zeichensatz	laden	^B oder ^C

nur Text-Modus:

Zeichen	setzen	' ' ... 'ß', DEL
Cursor	rechts	^I
Cursor	links	^H
Cursor	hoch	^K
Cursor	tief	^J
Cursor	an den Zeilenanfang	^M oder <CR>
Text-Modus verlassen		^ü oder ^ö

Anhang C:

ASCII-Tabelle

binär	hex	dez	ASCII	binär	hex	dez	ASCII	binär	hex	dez	ASCII	binär	hex	dez	ASCII
00000000	00	0	NUL	00100000	20	32	SPACE	01000000	40	64	§	01100000	60	96	`
00000001	01	1	SOH	00100001	21	33	!	01000001	41	65	A	01100001	61	97	a
00000010	02	2	STX	00100010	22	34	"	01000010	42	66	B	01100010	62	98	b
00000011	03	3	ETX	00100011	23	35	#	01000011	43	67	C	01100011	63	99	c
00000100	04	4	EOT	00100100	24	36	$	01000100	44	68	D	01100100	64	100	d
00000101	05	5	ENQ	00100101	25	37	%	01000101	45	69	E	01100101	65	101	e
00000110	06	6	ACK	00100110	26	38	&	01000110	46	70	F	01100110	66	102	f
00000111	07	7	BEL	00100111	27	39	'	01000111	47	71	G	01100111	67	103	g
00001000	08	8	BS	00101000	28	40	(	01001000	48	72	H	01101000	68	104	h
00001001	09	9	HT	00101001	29	41	)	01001001	49	73	I	01101001	69	105	i
00001010	0A	10	LF	00101010	2A	42	*	01001010	4A	74	J	01101010	6A	106	j
00001011	0B	11	VT	00101011	2B	43	+	01001011	4B	75	K	01101011	6B	107	k
00001100	0C	12	FF	00101100	2C	44	,	01001100	4C	76	L	01101100	6C	108	l
00001101	0D	13	CR	00101101	2D	45	-	01001101	4D	77	M	01101101	6D	109	m
00001110	0E	14	SO	00101110	2E	46	.	01001110	4E	78	N	01101110	6E	110	n
00001111	0F	15	SI	00101111	2F	47	/	01001111	4F	79	O	01101111	6F	111	o
00010000	10	16	DLE	00110000	30	48	0	01010000	50	80	P	01110000	70	112	p
00010001	11	17	DC1	00110001	31	49	1	01010001	51	81	Q	01110001	71	113	q
00010010	12	18	DC2	00110010	32	50	2	01010010	52	82	R	01110010	72	114	r
00010011	13	19	DC3	00110011	33	51	3	01010011	53	83	S	01110011	73	115	s
00010100	14	20	DC4	00110100	34	52	4	01010100	54	84	T	01110100	74	116	t
00010101	15	21	NAK	00110101	35	53	5	01010101	55	85	U	01110101	75	117	u
00010110	16	22	SYN	00110110	36	54	6	01010110	56	86	V	01110110	76	118	v
00010111	17	23	ETB	00110111	37	55	7	01010111	57	87	W	01110111	77	119	w
00011000	18	24	CAN	00111000	38	56	8	01011000	58	88	X	01111000	78	120	x
00011001	19	25	EM	00111001	39	57	9	01011001	59	89	Y	01111001	79	121	y
00011010	1A	26	SUB	00111010	3A	58	:	01011010	5A	90	Z	01111010	7A	122	z
00011011	1B	27	ESC	00111011	3B	59	;	01011011	5B	91	Ä	01111011	7B	123	ä
00011100	1C	28	FS	00111100	3C	60	<	01011100	5C	92	Ö	01111100	7C	124	ö
00011101	1D	29	GS	00111101	3D	61	=	01011101	5D	93	Ü	01111101	7D	125	ü
00011110	1E	30	RS	00111110	3E	62	>	01011110	5E	94	^	01111110	7E	126	ß
00011111	1F	31	US	00111111	3F	63	?	01011111	5F	95	_	01111111	7F	127	DEL

Diese Tabelle zeigt die deutsche Version des ASCII-Codes!

Anhang D:

Listing der Steuertabelle für Tastaturfunktionen

```
                        C       ;-------------------------------------------------------
                        C       ;       KEYBOARD        CONTROL CHARACTERS
                        C       ;                       BIT 7 MUST BE ON
                        C       ;-------------------------------------------------------
                        C
                        C       ;****** 1 BYTE CONTROL ******
                        C
19A0                    C       KTAB1:  REPT    32-KEYNTR
                        C               DEFB    0                       ; RESERVED
                        C               ENDM
19A0    00              C+              DEFB    0                       ; RESERVED
19A1    00              C+              DEFB    0                       ; RESERVED
19A2    00              C+              DEFB    0                       ; RESERVED
19A3    00              C+              DEFB    0                       ; RESERVED
19A4    00              C               DEFB    0                       ; RESET VIDEO I (COLDSTART)
19A5    00              C               DEFB    0                       ;  80 CHARACTERS PER LINE (25 LINES PER SCREEN)
19A6    00              C               DEFB    0                       ;  96 CHARACTERS PER LINE (25 LINES PER SCREEN)
19A7    00              C               DEFB    0                       ; 132 CHARACTERS PER LINE (25 LINES PER SCREEN)
19A8    85              C               DEFB    85H                     ; SCROLL UP
19A9    86              C               DEFB    86H                     ; SCROLL DOWN
19AA    00              C               DEFB    0                       ; CURSOR ON
19AB    00              C               DEFB    0                       ; CURSOR BLINK ON
19AC    00 00           C               DEFB    0 ,0                    ; INVERS SCREEN ON/OFF
19AE    00              C               DEFB    0                       ; DISPLAY TEST PATTERN
19AF    00              C               DEFB    0                       ; GERMAN CHARACTER SET
19B0    00              C               DEFB    0                       ; US CHARACTER SET
19B1    00              C               DEFB    0                       ;  80 CHARACTERS PER LINE (24 LINES PER SCREEN)
19B2    00              C               DEFB    0                       ;  96 CHARACTERS PER LINE (24 LINES PER SCREEN)
19B3    00              C               DEFB    0                       ; 132 CHARACTERS PER LINE (24 LINES PER SCREEN)
19B4    00              C               DEFB    0                       ; ENABLE TERMINAL 1
19B5    00              C               DEFB    0                       ; ENABLE TERMINAL 2
19B6    00              C               DEFB    0                       ; ENABLE TERMINAL 3
19B7    00              C               DEFB    0                       ; ENABLE TERMINAL 4
19B8    81 82           C               DEFB    81H, 82H                ; PAGE  UP/DOWN
19BA    83              C               DEFB    83H                     ; ACTUAL PAGE
19BB    00              C               DEFB    0                       ; KEYBOARD BUFFER ACTIV
19BC    00              C               DEFB    0                       ; KEYBOARD BUFFER NOT ACTIV
19BD    84              C               DEFB    84H                     ; CLEAR KEYBOARD BUFFER
19BE    00              C               DEFB    0                       ; 3 BYTE SEQUENCE
19BF    80              C       K1BYTE: DEFB    80H                     ; 2 ODER 3 BYTE SEQUENCE
                        C
                        C       ;****** 2 BYTE CONTROL ******
                        C
19C0                    C       KTAB2:  REPT    32-KEYNTR
                        C               DEFB    0                       ; RESERVED
                        C               ENDM
19C0    00              C+              DEFB    0                       ; RESERVED
19C1    00              C+              DEFB    0                       ; RESERVED
19C2    00              C+              DEFB    0                       ; RESERVED
19C3    00              C+              DEFB    0                       ; RESERVED
19C4    00              C               DEFB    0                       ; RESET VIDEO I (COLDSTART)
19C5    00              C               DEFB    0                       ;  80 CHARACTERS PER LINE (25 LINES PER SCREEN)
19C6    00              C               DEFB    0                       ;  96 CHARACTERS PER LINE (25 LINES PER SCREEN)
19C7    00              C               DEFB    0                       ; 132 CHARACTERS PER LINE (25 LINES PER SCREEN)
19C8    00              C               DEFB    0                       ; SCROLL UP
19C9    00              C               DEFB    0                       ; SCROLL DOWN
19CA    00              C               DEFB    0                       ; CURSOR ON
19CB    88              C               DEFB    88H                     ; CURSOR BLINK ON
19CC    00 87           C               DEFB    0 ,87H                  ; INVERS SCREEN ON/OFF
19CE    00              C               DEFB    0                       ; DISPLAY TEST PATTERN
19CF    85              C               DEFB    85H                     ; GERMAN CHARACTER SET
19D0    00              C               DEFB    0                       ; US CHARACTER SET
```

```
19D1  00          C           DEFB   0                ;  80 CHARACTERS PER LINE (24 LINES PER SCREEN)
19D2  00          C           DEFB   0                ;  96 CHARACTERS PER LINE (24 LINES PER SCREEN)
19D3  00          C           DEFB   0                ; 132 CHARACTERS PER LINE (24 LINES PER SCREEN)
19D4  81          C           DEFB   81H              ; ENABLE TERMINAL 1
19D5  82          C           DEFB   82H              ; ENABLE TERMINAL 2
19D6  83          C           DEFB   83H              ; ENABLE TERMINAL 3
19D7  84          C           DEFB   84H              ; ENABLE TERMINAL 4
19D8  00 00       C           DEFB   0, 0             ; PAGE  UP/DOWN
19DA  00          C           DEFB   0                ; ACTUAL PAGE
19DB  86          C           DEFB   86H              ; KEYBOARD BUFFER ACTIV
19DC  00          C           DEFB   0                ; KEYBOARD BUFFER NOT ACTIV
19DD  00          C           DEFB   0                ; CLEAR KEYBOARD BUFFER
19DE  80          C           DEFB   80H              ; 3 BYTE SEQUENCE
19DF  00          C     K2BYTE: DEFB   0              ; 2 ODER 3 BYTE SEQUENCE
                  C
                  C     ;****** 3 BYTE CONTROL ******
                  C
19E0              C     KTAB3:  REPT   32-KEYNTR
                  C             DEFB   0              ; RESERVED
                  C             ENDM
19E0  00          C+            DEFB   0              ; RESERVED
19E1  00          C+            DEFB   0              ; RESERVED
19E2  00          C+            DEFB   0              ; RESERVED
19E3  00          C+            DEFB   0              ; RESERVED
19E4  8C          C             DEFB   8CH            ; RESET VIDEO I (COLDSTART)
19E5  89          C             DEFB   89H            ;  80 CHARACTERS PER LINE (25 LINES PER SCREEN)
19E6  8A          C             DEFB   8AH            ;  96 CHARACTERS PER LINE (25 LINES PER SCREEN)
19E7  8B          C             DEFB   8BH            ; 132 CHARACTERS PER LINE (25 LINES PER SCREEN)
19E8  00          C             DEFB   0              ; SCROLL UP
19E9  00          C             DEFB   0              ; SCROLL DOWN
19EA  88          C             DEFB   88H            ; CURSOR ON
19EB  00          C             DEFB   0              ; CURSOR BLINK ON
19EC  87 00       C             DEFB   87H ,0         ; INVERS SCREEN ON/OFF
19EE  84          C             DEFB   84H            ; DISPLAY TEST PATTERN
19EF  00          C             DEFB   0              ; GERMAN CHARACTER SET
19F0  85          C             DEFB   85H            ; US CHARACTER SET
19F1  81          C             DEFB   81H            ;  80 CHARACTERS PER LINE (24 LINES PER SCREEN)
19F2  82          C             DEFB   82H            ;  96 CHARACTERS PER LINE (24 LINES PER SCREEN)
19F3  83          C             DEFB   83H            ; 132 CHARACTERS PER LINE (24 LINES PER SCREEN)
19F4  00          C             DEFB   0              ; ENABLE TERMINAL 1
19F5  00          C             DEFB   0              ; ENABLE TERMINAL 2
19F6  00          C             DEFB   0              ; ENABLE TERMINAL 3
19F7  00          C             DEFB   0              ; ENABLE TERMINAL 4
19F8  00 00       C             DEFB   0, 0           ; PAGE  UP/DOWN
19FA  00          C             DEFB   0              ; ACTUAL PAGE
19FB  00          C             DEFB   0              ; KEYBOARD BUFFER ACTIV
19FC  86          C             DEFB   86H            ; KEYBOARD BUFFER NOT ACTIV
19FD  00          C             DEFB   0              ; CLEAR KEYBOARD BUFFER
19FE  00          C             DEFB   0              ; 3 BYTE SEQUENCE
19FF  00          C     K3BYTE: DEFB   0              ; 2 ODER 3 BYTE SEQUENCE
                  C
                  C
```

Anhang E:

Listing der Terminal-Emulationstabellen

```
                        C       ;-----------------------------------------------------------
                        C       ;       TERMINAL 1 CONTROL CHARACTERS (LIKE ADDS VIEWPOINT)
                        C       ;                                       ++ ARE EXTENSIONS
                        C       ;-----------------------------------------------------------
1A00                    C       TABLE1: REPT    128-ENTRIES
                        C               DEFB    0                       ;RESERVED
                        C               ENDM
1A00    00              C+              DEFB    0                       ;RESERVED
1A01    00              C+              DEFB    0                       ;RESERVED
1A02    00              C+              DEFB    0                       ;RESERVED
1A03    00              C+              DEFB    0                       ;RESERVED
1A04    00              C+              DEFB    0                       ;RESERVED
1A05    00              C+              DEFB    0                       ;RESERVED
1A06    00              C+              DEFB    0                       ;RESERVED
1A07    00              C+              DEFB    0                       ;RESERVED
1A08    00              C+              DEFB    0                       ;RESERVED
1A09    00              C+              DEFB    0                       ;RESERVED
1A0A    00              C+              DEFB    0                       ;RESERVED
1A0B    00              C+              DEFB    0                       ;RESERVED
1A0C    00              C+              DEFB    0                       ;RESERVED
1A0D    00              C               defb    0                       ;reset cursor stack
1A0E    00              C               defb    0                       ;swap cursor
1A0F    00 00           C               defb    0, 0                    ;pop/push cursor
1A11    00 00 00        C               DEFB    0, 0, 0                 ;40/48/ 66 CHARACTERS PER LINE (12 LINES PER SCREEN)
1A14    00 00 00        C               DEFB    0, 0, 0                 ;80/96/132 CHARACTERS PER LINE (12 LINES PER SCREEN)
1A17    00 00 00        C               DEFB    0, 0, 0                 ;40/48/ 66 CHARACTERS PER LINE (24 LINES PER SCREEN)
1A1A    00 00 00        C               DEFB    0, 0, 0                 ;80/96/132 CHARACTERS PER LINE (25 LINES PER SCREEN)
1A1D    00 00           C               DEFB    0, 0                    ;SCREEN DARK    ON/OFF
1A1F    00              C               DEFB    0                       ;CLEAR WINDOW
1A20    00 00           C               DEFB    0, 0                    ;WINDOW SCROLL  UP/DOWN
1A22    00              C               defb    0                       ;load 6845
1A23    00              C               DEFB    0                       ;IGNORE NEXT CHAR
1A24    00              C               DEFB    0                       ;DISPLAY TEST PATTERN
1A25    00              C               DEFB    0                       ;CLEAR KEYBOARD BUFFER
1A26    00 00           C               DEFB    0, 0                    ;KEYBOARD BUFFER ACTIVE/NOT ACTIV
1A28    07              C               DEFB    BEL                     ;RING BELL
1A29    18              C               DEFB    CAN                     ;CURSOR ON
1A2A    00              C               DEFB    0                       ;CURSOR BLINK ON
1A2B    17              C               DEFB    ETB                     ;CURSOR OFF
1A2C    00 00           C               DEFB    0, 0                    ;KEYBOARD       UNLOCK/LOCK
1A2E    00 00           C               DEFB    0 ,0                    ;INVERS SCREEN  ON/OFF
1A30    00 00           C               DEFB    0, 0                    ;CHAR INVERS    ON/OFF
1A32    00 00           C               DEFB    0, 0                    ;BLINKING CHAR  ON/OFF
1A34    00 00           C               DEFB    0, 0                    ;UNDERLINE      ON/OFF
1A36    00 00           C               DEFB    0, 0                    ;HALF INTENSITY ON/OFF
1A38    0F              C               DEFB    SI                      ;NORMAL VIDEO
1A39    00              C               DEFB    0                       ;CHAR INVERS ONLY
1A3A    00              C               DEFB    0                       ;BLINK ONLY
1A3B    00              C               DEFB    0                       ;UNDERLINE ONLY
1A3C    00              C               DEFB    0                       ;HALF INTENSITY ONLY
1A3D    00              C               DEFB    0                       ;HALF INTENSITY BLINK
1A3E    0E              C               DEFB    SO                      ;CHAR INVERS + HALF INTENSITY
1A3F    00              C               DEFB    0                       ;CHAR INVERS + BLINK
1A40    00              C               DEFB    0                       ;CHAR INVERS + HALF INT. + BLINK
1A41    00              C               DEFB    0                       ;UNDERLINE + HALF INTENSITY
1A42    00              C               DEFB    0                       ;UNDERLINE BLINK
1A43    00              C               DEFB    0                       ;UNDERLINE + HALF INT. + BLINK
1A44    00              C               DEFB    0                       ;IDENTIFICATION
1A45    00              C               DEFB    0                       ;LOAD STRING
1A46    00              C               DEFB    0                       ;CLEAR STRING BUFFER
1A47    0C              C               DEFB    FF                      ;FORMFEED
1A48    00              C               DEFB    0                       ;CHANGE CHARACTER CODE SET (KEYBOARD)
```

```
1A49  00           C            DEFB   0                ;CHANGE CHARACTER CODE SET (SCREEN)
1A4A  00 00        C            DEFB   0, 0             ;GERMAN/US CHARACTER SET
1A4C  00 00 00     C            DEFB   0, 0, 0          ;80/96/132 CHARACTERS PER LINE (24 LINES PER SCREEN)
1A4F  00           C            DEFB   0                ;ENABLE TERMINAL1
1A50  00           C            DEFB   0                ;ENABLE TERMINAL2
1A51  00           C            DEFB   0                ;ENABLE TERMINAL3
1A52  00           C            DEFB   0                ;ENABLE TERMINAL4
1A53  00           C            DEFB   0                ;ENABLE TERMINAL5
1A54  00           C            DEFB   0                ;LOAD   TERMINAL5
1A55  00           C            DEFB   0                ;GRAFIC MODE 1
1A56  00           C            DEFB   0                ;GRAFIC MODE 2
1A57  00           C            DEFB   0                ;RESET VIDEO I (COLDSTART)
1A58  00           C            DEFB   0                ;ERASE TO  END  OF SCREEN
1A59  00           C            DEFB   0                ;ERASE TO START OF SCREEN
1A5A  12           C            DEFB   DC2              ;ERASE SCREEN                    ++
1A5B  13           C            DEFB   DC3              ;ERASE SCREEN + CURSOR HOME      ++
1A5C  01           C            DEFB   SOH              ;CURSOR HOME
1A5D  00           C            DEFB   0                ;ERASE TO END OF LINE
1A5E  00           C            DEFB   0                ;ERASE LINE
1A5F  00 00        C            DEFB   0, 0             ;PAGE           UP/DOWN
1A61  00           C            DEFB   0                ;PAGE           ACTUAL
1A62  00 00        C            DEFB   0, 0             ;SCROLL         UP/DOWN
1A64  00 00        C            DEFB   0, 0             ;PARTIAL SCROLL UP/DOWN
1A66  00           C            DEFB   0                ;SEND CURSOR ADDRESS WITHOUT OFFSET => X ; Y
1A67  00           C            DEFB   0                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H => X ; Y
1A68  00           C            DEFB   0                ;SEND CURSOR ADDRESS WITHOUT OFFSET => Y ; X
1A69  00           C            DEFB   0                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H => Y ; X
1A6A  00           C            DEFB   0                ;SEND CHAR AT CURSOR
1A6B  06           C            DEFB   ACK              ;CURSOR FORWARD
1A6C  00 15        C            DEFB   0, NAK           ;CURSOR RIGHT/LEFT
1A6E  1A 00        C            DEFB   SUB, 0           ;CURSOR UP/DOWN
1A70  00 00        C            DEFB   0, 0             ;RESERVED: DO NOT USE
1A72  0B           C            DEFB   VT               ;VERTICAL   CURSOR ADDRESSING
1A73  10           C            DEFB   DLE              ;HORIZONTAL CURSOR ADDRESSING
1A74  00           C            DEFB   0                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET => X ; Y
1A75  00           C            DEFB   0                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H => X ; Y
1A76  00           C            DEFB   0                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET => Y ; X
1A77  00           C            DEFB   0                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H => Y ; X
1A78  00           C            DEFB   0                ;CARRIAGE RETURN + ERASE END OF LINE
1A79  00           C            DEFB   0                ;CARRIAGE RETURN + LINE FEED
1A7A  0D           C            DEFB   CR               ;CARRIAGE RETURN
1A7B  00           C            DEFB   0                ;REVERSE LINEFEED
1A7C  0A           C            DEFB   LF               ;LINEFEED
1A7D  08           C            DEFB   BS               ;BACKSPACE
1A7E  00           C            DEFB   0                ;TWO BYTE SEQUENCE
1A7F  1B           C    TICON:  DEFB   ESC              ;BYTE (ESCAPE) SEQUENCE
                   C            PAGE
```

```
                        C
                        C       ;-----------------------------------------------------------
                        C       ;       TERMINAL 1      BYTE (ESC) - SEQUENCES
                        C       ;-----------------------------------------------------------
                        C
                        C               REPT    128-ENTRIES
                        C               DEFB    0                       ;RESERVED
                        C               ENDM
1A80    00              C+              DEFB    0                       ;RESERVED
1A81    00              C+              DEFB    0                       ;RESERVED
1A82    00              C+              DEFB    0                       ;RESERVED
1A83    00              C+              DEFB    0                       ;RESERVED
1A84    00              C+              DEFB    0                       ;RESERVED
1A85    00              C+              DEFB    0                       ;RESERVED
1A86    00              C+              DEFB    0                       ;RESERVED
1A87    00              C+              DEFB    0                       ;RESERVED
1A88    00              C+              DEFB    0                       ;RESERVED
1A89    00              C+              DEFB    0                       ;RESERVED
1A8A    00              C+              DEFB    0                       ;RESERVED
1A8B    00              C+              DEFB    0                       ;RESERVED
1A8C    00              C+              DEFB    0                       ;RESERVED
1A8D    00              C               defb    0                       ;reset cursor stack
1A8E    00              C               defb    0                       ;swap cursor
1A8F    00 00           C               defb    0, 0                    ;pop/push cursor
1A91    00 00 00        C               DEFB    0, 0, 0                 ;40/48/ 66 CHARACTERS PER LINE (12 LINES PER SCREEN)
1A94    00 00 00        C               DEFB    0, 0, 0                 ;80/96/132 CHARACTERS PER LINE (12 LINES PER SCREEN)
1A97    00 00 00        C               DEFB    0, 0, 0                 ;40/48/ 66 CHARACTERS PER LINE (24 LINES PER SCREEN)
1A9A    00 00 00        C               DEFB    0, 0, 0                 ;80/96/132 CHARACTERS PER LINE (25 LINES PER SCREEN)
1A9D    00 00           C               DEFB    0, 0                    ;SCREEN DARK    ON/OFF
1A9F    00              C               DEFB    0                       ;CLEAR WINDOW
1AA0    00 00           C               DEFB    0, 0                    ;WINDOW SCROLL  UP/DOWN
1AA2    00              C               defb    0                       ;load 6845
1AA3    30              C               DEFB    '0'                     ;IGNORE NEXT CHAR
1AA4    00              C               DEFB    0                       ;DISPLAY TEST   PATTERN
1AA5    00              C               DEFB    0                       ;CLEAR KEYBOARD BUFFER
1AA6    00 00           C               DEFB    0, 0                    ;KEYBOARD BUFFER ACTIVE/NOT ACTIV
1AA8    00              C               DEFB    0                       ;RING BELL
1AA9    43              C               DEFB    'C'                     ;CURSOR ON                      ++
1AAA    63              C               DEFB    'c'                     ;CURSOR BLINK ON                ++
1AAB    44              C               DEFB    'D'                     ;CURSOR OFF                     ++
1AAC    36 35           C               DEFB    '6', '5'                ;KEYBOARD       UNLOCK/LOCK
1AAE    00 00           C               DEFB    0, 0                    ;INVERS SCREEN  ON/OFF
1AB0    00 00           C               DEFB    0, 0                    ;CHAR INVERS    ON/OFF
1AB2    00 00           C               DEFB    0, 0                    ;BLINKING CHAR  ON/OFF
1AB4    00 00           C               DEFB    0, 0                    ;UNDERLINE      ON/OFF
1AB6    00 00           C               DEFB    0, 0                    ;HALF INTENSITY ON/OFF
1AB8    00              C               DEFB    0                       ;NORMAL VIDEO
1AB9    00              C               DEFB    0                       ;CHAR INVERS ONLY
1ABA    00              C               DEFB    0                       ;BLINK ONLY
1ABB    00              C               DEFB    0                       ;UNDERLINE ONLY
1ABC    00              C               DEFB    0                       ;HALF INTENSITY ONLY
1ABD    00              C               DEFB    0                       ;HALF INTENSITY BLINK
1ABE    00              C               DEFB    0                       ;CHAR INVERS + HALF INTENSITY
1ABF    00              C               DEFB    0                       ;CHAR INVERS + BLINK
1AC0    00              C               DEFB    0                       ;CHAR INVERS + HALF INT. + BLINK
1AC1    00              C               DEFB    0                       ;UNDERLINE + HALF INTENSITY
1AC2    00              C               DEFB    0                       ;UNDERLINE BLINK
1AC3    00              C               DEFB    0                       ;UNDERLINE + HALF INT. + BLINK
1AC4    FF              C               DEFB    0FFH                    ;IDENTIFICATION                 ++
1AC5    61              C               DEFB    'a'                     ;LOAD STRING                    ++
1AC6    62              C               DEFB    'b'                     ;CLEAR STRING BUFFER            ++
```

```
1AC7  00           C           DEFB  0                ;FORMFEED
1AC8  00           C           DEFB  0                ;CHANGE CHARACTER CODE SET (KEYBOARD)
1AC9  00           C           DEFB  0                ;CHANGE CHARACTER CODE SET (SCREEN)
1ACA  00 00        C           DEFB  0, 0             ;GERMAN/US CHARACTER SET
1ACC  38 39 37     C           DEFB  '8', '9', '7'    ;80/96/132 CHARACTERS PER LINE  ++
1ACF  71           C           DEFB  'q'              ;ENABLE TERMINAL1               ++
1AD0  72           C           DEFB  'r'              ;ENABLE TERMINAL2               ++
1AD1  73           C           DEFB  's'              ;ENABLE TERMINAL3               ++
1AD2  74           C           DEFB  't'              ;ENABLE TERMINAL4               ++
1AD3  75           C           DEFB  'u'              ;ENABLE TERMINAL5               ++
1AD4  76           C           DEFB  'v'              ;LOAD   TERMINAL5               ++
1AD5  00           C           DEFB  0                ;GRAFIC MODE 1
1AD6  00           C           DEFB  0                ;GRAFIC MODE 2
1AD7  00           C           DEFB  0                ;RESET VIDEO I (COLDSTART)
1AD8  6B           C           DEFB  'k'              ;ERASE TO  END  OF SCREEN
1AD9  4C           C           DEFB  'L'              ;ERASE TO START OF SCREEN       ++
1ADA  00           C           DEFB  0                ;ERASE SCREEN
1ADB  00           C           DEFB  0                ;ERASE SCREEN + CURSOR HOME
1ADC  00           C           DEFB  0                ;CURSOR HOME
1ADD  4B           C           DEFB  'K'              ;ERASE TO END OF LINE
1ADE  00           C           DEFB  0                ;ERASE LINE
1ADF  45 46        C           DEFB  'E', 'F'         ;PAGE           UP/DOWN         ++
1AE1  47           C           DEFB  'G'              ;PAGE           ACTUAL          ++
1AE2  00 00        C           DEFB  0, 0             ;SCROLL         UP/DOWN
1AE4  00 00        C           DEFB  0, 0             ;PARTIAL SCROLL UP/DOWN
1AE6  00           C           DEFB  0                ;SEND CURSOR ADDRESS WITHOUT OFFSET =) X ; Y
1AE7  00           C           DEFB  0                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H =) X ; Y
1AE8  00           C           DEFB  0                ;SEND CURSOR ADDRESS WITHOUT OFFSET =) Y ; X
1AE9  00           C           DEFB  0                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H =) Y ; X
1AEA  00           C           DEFB  0                ;SEND CHAR AT CURSOR
1AEB  00           C           DEFB  0                ;CURSOR FORWARD
1AEC  00 00        C           DEFB  0, 0             ;CURSOR RIGHT/LEFT
1AEE  00 00        C           DEFB  0, 0             ;CURSOR UP/DOWN
1AF0  00 00        C           DEFB  0, 0             ;RESERVED: DO NOT USE
1AF2  00           C           DEFB  0                ;VERTICAL   CURSOR ADDRESSING
1AF3  00           C           DEFB  0                ;HORIZONTAL CURSOR ADDRESSING
1AF4  00           C           DEFB  0                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET =) X ; Y
1AF5  00           C           DEFB  0                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H =) X ; Y
1AF6  00           C           DEFB  0                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET =) Y ; X
1AF7  59           C           DEFB  'Y'              ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H =) Y ; X
1AF8  00           C           DEFB  0                ;CARRIAGE RETURN + ERASE END OF LINE
1AF9  00           C           DEFB  0                ;CARRIAGE RETURN + LINE FEED
1AFA  00           C           DEFB  0                ;CARRIAGE RETURN
1AFB  00           C           DEFB  0                ;REVERSE LINEFEED
1AFC  00           C           DEFB  0                ;LINEFEED
1AFD  00           C           DEFB  0                ;BACKSPACE
1AFE  58           C           DEFB  'X'              ;TWO BYTE SEQUENCE              ++
1AFF  00           C  T1ESC:   DEFB  0                ;BYTE (ESCAPE) SEQUENCE
                   C           PAGE
```

```
                        C
                        C       ;-----------------------------------------------------------
                        C       ;       TERMINAL 1      TWO BYTE SEQUENCES, (ESC x) - SEQUENCES
                        C       ;                       NOT COMPATIBLE WITH ADDS VIEWPOINT
                        C       ;-----------------------------------------------------------
                        C               REPT    128-ENTRIES
                        C               DEFB    0                       ;RESERVED
                        C               ENDM
1B00    00              C+              DEFB    0                       ;RESERVED
1B01    00              C+              DEFB    0                       ;RESERVED
1B02    00              C+              DEFB    0                       ;RESERVED
1B03    00              C+              DEFB    0                       ;RESERVED
1B04    00              C+              DEFB    0                       ;RESERVED
1B05    00              C+              DEFB    0                       ;RESERVED
1B06    00              C+              DEFB    0                       ;RESERVED
1B07    00              C+              DEFB    0                       ;RESERVED
1B08    00              C+              DEFB    0                       ;RESERVED
1B09    00              C+              DEFB    0                       ;RESERVED
1B0A    00              C+              DEFB    0                       ;RESERVED
1B0B    00              C+              DEFB    0                       ;RESERVED
1B0C    00              C+              DEFB    0                       ;RESERVED
1B0D    73              C               defb    115                     ;reset cursor stack
1B0E    72              C               defb    114                     ;swap cursor
1B0F    71 70           C               defb    113, 112                ;pop/push cursor
1B11    6F 6E 6D        C               DEFB    111,110,109             ;40/48/ 66 CHARACTERS PER LINE (12 LINES PER SCREEN)
1B14    6C 6B 6A        C               DEFB    108,107,106             ;80/96/132 CHARACTERS PER LINE (12 LINES PER SCREEN)
1B17    69 68 67        C               DEFB    105,104,103             ;40/48/ 66 CHARACTERS PER LINE (24 LINES PER SCREEN)
1B1A    66 65 64        C               DEFB    102,101,100             ;80/96/132 CHARACTERS PER LINE (25 LINES PER SCREEN)
1B1D    63 62           C               DEFB     99, 98                 ;SCREEN DARK    ON/OFF
1B1F    61              C               DEFB     97                     ;CLEAR WINDOW
1B20    60 5F           C               DEFB     96, 95                 ;WINDOW SCROLL  UP/DOWN
1B22    5E              C               defb     94                     ;load 6845
1B23    5D              C               DEFB     93                     ;IGNORE NEXT CHAR
1B24    5C              C               DEFB     92                     ;DISPLAY TEST PATTERN
1B25    5B              C               DEFB     91                     ;CLEAR KEYBOARD BUFFER
1B26    5A 59           C               DEFB     90, 89                 ;KEYBOARD BUFFER ACTIVE/NOT ACTIV
1B28    58              C               DEFB     88                     ;RING BELL
1B29    57              C               DEFB     87                     ;CURSOR ON
1B2A    56              C               DEFB     86                     ;CURSOR BLINK ON
1B2B    55              C               DEFB     85                     ;CURSOR OFF
1B2C    54 53           C               DEFB     84, 83                 ;KEYBOARD       UNLOCK/LOCK
1B2E    52 51           C               DEFB     82, 81                 ;INVERS SCREEN  ON/OFF
1B30    50 4F           C               DEFB     80, 79                 ;CHAR INVERS    ON/OFF
1B32    4E 4D           C               DEFB     78, 77                 ;BLINKING CHAR  ON/OFF
1B34    4C 4B           C               DEFB     76, 75                 ;UNDERLINE      ON/OFF
1B36    4A 49           C               DEFB     74, 73                 ;HALF INTENSITY ON/OFF
1B38    48              C               DEFB     72                     ;NORMAL VIDEO
1B39    47              C               DEFB     71                     ;CHAR INVERS ONLY
1B3A    46              C               DEFB     70                     ;BLINK ONLY
1B3B    45              C               DEFB     69                     ;UNDERLINE ONLY
1B3C    44              C               DEFB     68                     ;HALF INTENSITY ONLY
1B3D    43              C               DEFB     67                     ;HALF INTENSITY BLINK
1B3E    42              C               DEFB     66                     ;CHAR INVERS + HALF INTENSITY
1B3F    41              C               DEFB     65                     ;CHAR INVERS + BLINK
1B40    40              C               DEFB     64                     ;CHAR INVERS + HALF INT. + BLINK
1B41    3F              C               DEFB     63                     ;UNDERLINE + HALF INTENSITY
1B42    3E              C               DEFB     62                     ;UNDERLINE BLINK
1B43    3D              C               DEFB     61                     ;UNDERLINE + HALF INT. + BLINK
1B44    3C              C               DEFB     60                     ;IDENTIFICATION
1B45    3B              C               DEFB     59                     ;LOAD STRING
1B46    3A              C               DEFB     58                     ;CLEAR STRING BUFFER
```

```
1B47  39           C          DEFB   57              ;FORMFEED
1B48  38           C          DEFB   56              ;CHANGE CHARACTER CODE SET (KEYBOARD)
1B49  37           C          DEFB   55              ;CHANGE CHARACTER CODE SET (SCREEN)
1B4A  36 35        C          DEFB   54, 53          ;GERMAN/US CHARACTER SET
1B4C  34 33 32     C          DEFB   52, 51, 50      ;80/96/132 CHARACTERS PER LINE (24 LINES PER SCREEN)
1B4F  31           C          DEFB   49              ;ENABLE TERMINAL1
1B50  30           C          DEFB   48              ;ENABLE TERMINAL2
1B51  2F           C          DEFB   47              ;ENABLE TERMINAL3
1B52  2E           C          DEFB   46              ;ENABLE TERMINAL4
1B53  2D           C          DEFB   45              ;ENABLE TERMINAL5
1B54  2C           C          DEFB   44              ;LOAD   TERMINAL5
1B55  2B           C          DEFB   43              ;GRAFIC MODE 1
1B56  2A           C          DEFB   42              ;GRAFIC MODE 2
1B57  29           C          DEFB   41              ;RESET VIDEO I (COLDSTART)
1B58  28           C          DEFB   40              ;ERASE TO  END  OF SCREEN
1B59  27           C          DEFB   39              ;ERASE TO START OF SCREEN
1B5A  26           C          DEFB   38              ;ERASE SCREEN
1B5B  25           C          DEFB   37              ;ERASE SCREEN + CURSOR HOME
1B5C  24           C          DEFB   36              ;CURSOR HOME
1B5D  23           C          DEFB   35              ;ERASE TO END OF LINE
1B5E  22           C          DEFB   34              ;ERASE LINE
1B5F  21 20        C          DEFB   33, 32          ;PAGE           UP/DOWN
1B61  1F           C          DEFB   31              ;PAGE           ACTUAL
1B62  1E 1D        C          DEFB   30, 29          ;SCROLL         UP/DOWN
1B64  1C 1B        C          DEFB   28, 27          ;PARTIAL SCROLL UP/DOWN
1B66  1A           C          DEFB   26              ;SEND CURSOR ADDRESS WITHOUT OFFSET ==) X ; Y
1B67  19           C          DEFB   25              ;SEND CURSOR ADDRESS WITH OFFSET OF 20H ==) X ; Y
1B68  18           C          DEFB   24              ;SEND CURSOR ADDRESS WITHOUT OFFSET ==) Y ; X
1B69  17           C          DEFB   23              ;SEND CURSOR ADDRESS WITH OFFSET OF 20H ==) Y ; X
1B6A  16           C          DEFB   22              ;SEND CHAR AT CURSOR
1B6B  15           C          DEFB   21              ;CURSOR FORWARD
1B6C  14 13        C          DEFB   20, 19          ;CURSOR RIGHT/LEFT
1B6E  12 11        C          DEFB   18, 17          ;CURSOR UP/DOWN
1B70  10 0F        C          DEFB   16, 15          ;RESERVED: DO NOT USE
1B72  0E           C          DEFB   14              ;VERTICAL   CURSOR ADDRESSING
1B73  0D           C          DEFB   13              ;HORIZONTAL CURSOR ADDRESSING
1B74  0C           C          DEFB   12              ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET ==) X ; Y
1B75  0B           C          DEFB   11              ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H ==) X ; Y
1B76  0A           C          DEFB   10              ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET ==) Y ; X
1B77  09           C          DEFB    9              ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H ==) Y ; X
1B78  08           C          DEFB    8              ;CARRIAGE RETURN + ERASE END OF LINE
1B79  07           C          DEFB    7              ;CARRIAGE RETURN + LINE FEED
1B7A  06           C          DEFB    6              ;CARRIAGE RETURN
1B7B  05           C          DEFB    5              ;REVERSE LINEFEED
1B7C  04           C          DEFB    4              ;LINEFEED
1B7D  03           C          DEFB    3              ;BACKSPACE
1B7E  00           C          DEFB   0;2             ;TWO BYTE SEQUENCE
1B7F  00           C  T1ESCX: DEFB   0;1             ;BYTE (ESCAPE) SEQUENCE
```

```
                        C       ;-------------------------------------------------------------
                        C       ;       TERMINAL 2 (LIKE GEVEKE VISA 30/40)
                        C       ;                  ++ ARE EXTENSIONS
                        C       ;-------------------------------------------------------------
1B80                    C       TABLE2: REPT    128-ENTRIES
                        C               DEFB    0                       ;RESERVED
                        C               ENDM
1B80    00              C+              DEFB    0                       ;RESERVED
1B81    00              C+              DEFB    0                       ;RESERVED
1B82    00              C+              DEFB    0                       ;RESERVED
1B83    00              C+              DEFB    0                       ;RESERVED
1B84    00              C+              DEFB    0                       ;RESERVED
1B85    00              C+              DEFB    0                       ;RESERVED
1B86    00              C+              DEFB    0                       ;RESERVED
1B87    00              C+              DEFB    0                       ;RESERVED
1B88    00              C+              DEFB    0                       ;RESERVED
1B89    00              C+              DEFB    0                       ;RESERVED
1B8A    00              C+              DEFB    0                       ;RESERVED
1B8B    00              C+              DEFB    0                       ;RESERVED
1B8C    00              C+              DEFB    0                       ;RESERVED
1B8D    00              C               defb    0                       ;reset cursor stack
1B8E    00              C               defb    0                       ;swap cursor
1B8F    00 00           C               defb    0, 0                    ;pop/push cursor
1B91    00 00 00        C               DEFB    0, 0, 0                 ;40/48/ 66 CHARACTERS PER LINE (12 LINES PER SCREEN)
1B94    00 00 00        C               DEFB    0, 0, 0                 ;80/96/132 CHARACTERS PER LINE (12 LINES PER SCREEN)
1B97    00 00 00        C               DEFB    0, 0, 0                 ;40/48/ 66 CHARACTERS PER LINE (24 LINES PER SCREEN)
1B9A    00 00 00        C               DEFB    0, 0, 0                 ;80/96/132 CHARACTERS PER LINE (25 LINES PER SCREEN)
1B9D    00 00           C               DEFB    0, 0                    ;SCREEN DARK    ON/OFF
1B9F    00              C               DEFB    0                       ;CLEAR WINDOW
1BA0    00 00           C               DEFB    0, 0                    ;WINDOW SCROLL  UP/DOWN
1BA2    00              C               defb    0                       ;load 6845
1BA3    00              C               DEFB    0                       ;IGNORE NEXT CHAR
1BA4    00              C               DEFB    0                       ;DISPLAY TEST PATTERN
1BA5    00              C               DEFB    0                       ;CLEAR KEYBOARD BUFFER
1BA6    00 00           C               DEFB    0, 0                    ;KEYBOARD BUFFER ACTIVE/NOT ACTIV
1BA8    07              C               DEFB    BEL                     ;RING BELL
1BA9    00              C               DEFB    0                       ;CURSOR ON
1BAA    00              C               DEFB    0                       ;CURSOR BLINK ON
1BAB    00              C               DEFB    0                       ;CURSOR OFF
1BAC    00 00           C               DEFB    0, 0                    ;KEYBOARD       UNLOCK/LOCK
1BAE    00 00           C               DEFB    0 ,0                    ;INVERS SCREEN  ON/OFF
1BB0    00 00           C               DEFB    0, 0                    ;CHAR INVERS    ON/OFF
1BB2    00 00           C               DEFB    0, 0                    ;BLINKING CHAR  ON/OFF
1BB4    00 00           C               DEFB    0, 0                    ;UNDERLINE      ON/OFF
1BB6    00 00           C               DEFB    0, 0                    ;HALF INTENSITY ON/OFF
1BB8    00              C               DEFB    0                       ;NORMAL VIDEO
1BB9    00              C               DEFB    0                       ;CHAR INVERS ONLY
1BBA    00              C               DEFB    0                       ;BLINK ONLY
1BBB    00              C               DEFB    0                       ;UNDERLINE ONLY
1BBC    00              C               DEFB    0                       ;HALF INTENSITY ONLY
1BBD    00              C               DEFB    0                       ;HALF INTENSITY BLINK
1BBE    00              C               DEFB    0                       ;CHAR INVERS + HALF INTENSITY
1BBF    00              C               DEFB    0                       ;CHAR INVERS + BLINK
1BC0    00              C               DEFB    0                       ;CHAR INVERS + HALF INT. + BLINK
1BC1    00              C               DEFB    0                       ;UNDERLINE + HALF INTENSITY
1BC2    00              C               DEFB    0                       ;UNDERLINE BLINK
1BC3    00              C               DEFB    0                       ;UNDERLINE + HALF INT. + BLINK
1BC4    00              C               DEFB    0                       ;IDENTIFICATION
1BC5    00              C               DEFB    0                       ;LOAD STRING
1BC6    00              C               DEFB    0                       ;CLEAR STRING BUFFER
1BC7    00              C               DEFB    0                       ;FORMFEED
1BC8    00              C               DEFB    0                       ;CHANGE CHARACTER CODE SET (KEYBOARD)
```

```
1BC9  00           C            DEFB   0              ;CHANGE CHARACTER CODE SET (SCREEN)
1BCA  00 00        C            DEFB   0, 0           ;GERMAN/US CHARACTER SET
1BCC  00 00 00     C            DEFB   0, 0, 0        ;80/96/132 CHARACTERS PER LINE (24 LINES PER SCREEN)
1BCF  00           C            DEFB   0              ;ENABLE TERMINAL1
1BD0  00           C            DEFB   0              ;ENABLE TERMINAL2
1BD1  00           C            DEFB   0              ;ENABLE TERMINAL3
1BD2  00           C            DEFB   0              ;ENABLE TERMINAL4
1BD3  00           C            DEFB   0              ;ENABLE TERMINAL5
1BD4  00           C            DEFB   0              ;LOAD   TERMINAL5
1BD5  00           C            DEFB   0              ;GRAFIC MODE 1
1BD6  00           C            DEFB   0              ;GRAFIC MODE 2
1BD7  00           C            DEFB   0              ;RESET VIDEO I (COLDSTART)
1BD8  00           C            DEFB   0              ;ERASE TO  END  OF SCREEN
1BD9  00           C            DEFB   0              ;ERASE TO START OF SCREEN
1BDA  00           C            DEFB   0              ;ERASE SCREEN
1BDB  00           C            DEFB   0              ;ERASE SCREEN + CURSOR HOME
1BDC  00           C            DEFB   0              ;CURSOR HOME
1BDD  00           C            DEFB   0              ;ERASE TO END OF LINE
1BDE  00           C            DEFB   0              ;ERASE LINE
1BDF  00 00        C            DEFB   0, 0           ;PAGE           UP/DOWN
1BE1  00           C            DEFB   0              ;PAGE           ACTUAL
1BE2  00 00        C            DEFB   0, 0           ;SCROLL         UP/DOWN
1BE4  00 00        C            DEFB   0, 0           ;PARTIAL SCROLL UP/DOWN
1BE6  00           C            DEFB   0              ;SEND CURSOR ADDRESS WITHOUT OFFSET ==> X ; Y
1BE7  00           C            DEFB   0              ;SEND CURSOR ADDRESS WITH OFFSET OF 20H ==> X ; Y
1BE8  00           C            DEFB   0              ;SEND CURSOR ADDRESS WITHOUT OFFSET ==> Y ; X
1BE9  00           C            DEFB   0              ;SEND CURSOR ADDRESS WITH OFFSET OF 20H ==> Y ; X
1BEA  00           C            DEFB   0              ;SEND CHAR AT CURSOR
1BEB  00           C            DEFB   0              ;CURSOR FORWARD
1BEC  10 00        C            DEFB   DLE, 0         ;CURSOR RIGHT/LEFT
1BEE  00 00        C            DEFB   0, 0           ;CURSOR UP/DOWN
1BF0  00 00        C            DEFB   0, 0           ;RESERVED: DO NOT USE
1BF2  00           C            DEFB   0              ;VERTICAL   CURSOR ADDRESSING
1BF3  00           C            DEFB   0              ;HORIZONTAL CURSOR ADDRESSING
1BF4  00           C            DEFB   0              ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET ==> X ; Y
1BF5  00           C            DEFB   0              ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H ==> X ; Y
1BF6  00           C            DEFB   0              ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET ==> Y ; X
1BF7  00           C            DEFB   0              ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H ==> Y ; X
1BF8  00           C            DEFB   0              ;CARRIAGE RETURN + ERASE END OF LINE
1BF9  00           C            DEFB   0              ;CARRIAGE RETURN + LINE FEED
1BFA  0D           C            DEFB   CR             ;CARRIAGE RETURN
1BFB  00           C            DEFB   0              ;REVERSE LINEFEED
1BFC  0A           C            DEFB   LF             ;LINEFEED
1BFD  08           C            DEFB   BS             ;BACKSPACE
1BFE  00           C            DEFB   0              ;TWO BYTE SEQUENCE
1BFF  1B           C    T2CON:  DEFB   ESC            ;BYTE (ESCAPE) SEQUENCE
                   C            PAGE
```

```
                        C
                        C       ;-----------------------------------------------------------
                        C       ;       TERMINAL 2      BYTE (ESC) - SEQUENCES
                        C       ;-----------------------------------------------------------
                        C
                        C               REPT    128-ENTRIES
                        C               DEFB    0                       ;RESERVED
                        C               ENDM
1C00    00              C+              DEFB    0                       ;RESERVED
1C01    00              C+              DEFB    0                       ;RESERVED
1C02    00              C+              DEFB    0                       ;RESERVED
1C03    00              C+              DEFB    0                       ;RESERVED
1C04    00              C+              DEFB    0                       ;RESERVED
1C05    00              C+              DEFB    0                       ;RESERVED
1C06    00              C+              DEFB    0                       ;RESERVED
1C07    00              C+              DEFB    0                       ;RESERVED
1C08    00              C+              DEFB    0                       ;RESERVED
1C09    00              C+              DEFB    0                       ;RESERVED
1C0A    00              C+              DEFB    0                       ;RESERVED
1C0B    00              C+              DEFB    0                       ;RESERVED
1C0C    00              C+              DEFB    0                       ;RESERVED
1C0D    00              C               defb    0                       ;reset cursor stack
1C0E    00              C               defb    0                       ;swap cursor
1C0F    00 00           C               defb    0, 0                    ;pop/push cursor
1C11    00 00 00        C               DEFB    0, 0, 0                 ;40/48/ 66 CHARACTERS PER LINE (12 LINES PER SCREEN)
1C14    00 00 00        C               DEFB    0, 0, 0                 ;80/96/132 CHARACTERS PER LINE (12 LINES PER SCREEN)
1C17    00 00 00        C               DEFB    0, 0, 0                 ;40/48/ 66 CHARACTERS PER LINE (24 LINES PER SCREEN)
1C1A    00 00 00        C               DEFB    0, 0, 0                 ;80/96/132 CHARACTERS PER LINE (25 LINES PER SCREEN)
1C1D    00 00           C               DEFB    0, 0                    ;SCREEN DARK    ON/OFF
1C1F    00              C               DEFB    0                       ;CLEAR WINDOW
1C20    00 00           C               DEFB    0, 0                    ;WINDOW SCROLL  UP/DOWN
1C22    00              C               defb    0                       ;load 6845
1C23    00              C               DEFB    0                       ;IGNORE NEXT CHAR
1C24    22              C               DEFB    '"'                     ;DISPLAY TEST PATTERN
1C25    00              C               DEFB    0                       ;CLEAR KEYBOARD BUFFER
1C26    00 00           C               DEFB    0, 0                    ;KEYBOARD BUFFER ACTIVE/NOT ACTIV
1C28    00              C               DEFB    0                       ;RING BELL
1C29    00              C               DEFB    0                       ;CURSOR ON
1C2A    00              C               DEFB    0                       ;CURSOR BLINK ON
1C2B    00              C               DEFB    0                       ;CURSOR OFF
1C2C    06 15           C               DEFB    ACK, NAK                ;KEYBOARD       UNLOCK/LOCK
1C2E    00 00           C               DEFB    0, 0                    ;INVERS SCREEN  ON/OFF
1C30    00 00           C               DEFB    0, 0                    ;CHAR INVERS    ON/OFF
1C32    00 00           C               DEFB    0, 0                    ;BLINKING CHAR  ON/OFF
1C34    00 00           C               DEFB    0, 0                    ;UNDERLINE      ON/OFF
1C36    00 00           C               DEFB    0, 0                    ;HALF INTENSITY ON/OFF
1C38    1F              C               DEFB    US                      ;NORMAL VIDEO
1C39    00              C               DEFB    0                       ;CHAR INVERS ONLY
1C3A    00              C               DEFB    0                       ;BLINK ONLY
1C3B    00              C               DEFB    0                       ;UNDERLINE ONLY
1C3C    19              C               DEFB    EM                      ;HALF INTENSITY ONLY
1C3D    00              C               DEFB    0                       ;HALF INTENSITY BLINK
1C3E    00              C               DEFB    0                       ;CHAR INVERS + HALF INTENSITY
1C3F    00              C               DEFB    0                       ;CHAR INVERS + BLINK
1C40    00              C               DEFB    0                       ;CHAR INVERS + HALF INT. + BLINK
1C41    00              C               DEFB    0                       ;UNDERLINE + HALF INTENSITY
1C42    00              C               DEFB    0                       ;UNDERLINE BLINK
1C43    00              C               DEFB    0                       ;UNDERLINE + HALF INT. + BLINK
1C44    00              C               DEFB    0                       ;IDENTIFICATION
1C45    61              C               DEFB    'a'                     ;LOAD STRING                    ++
1C46    62              C               DEFB    'b'                     ;CLEAR STRING BUFFER            ++
```

```
1C47  00           C            DEFB   0                ;FORMFEED
1C48  00           C            DEFB   0                ;CHANGE CHARACTER CODE SET (KEYBOARD)
1C49  00           C            DEFB   0                ;CHANGE CHARACTER CODE SET (SCREEN)
1C4A  00 00        C            DEFB   0, 0             ;GERMAN/US CHARACTER SET
1C4C  38 39 37     C            DEFB   '8', '9', '7'    ;80/96/132 CHARACTERS PER LINE  ++
1C4F  71           C            DEFB   'q'              ;ENABLE TERMINAL1               ++
1C50  72           C            DEFB   'r'              ;ENABLE TERMINAL2               ++
1C51  73           C            DEFB   's'              ;ENABLE TERMINAL3               ++
1C52  74           C            DEFB   't'              ;ENABLE TERMINAL4               ++
1C53  75           C            DEFB   'u'              ;ENABLE TERMINAL5               ++
1C54  76           C            DEFB   'v'              ;LOAD   TERMINAL5               ++
1C55  00           C            DEFB   0                ;GRAFIC MODE 1
1C56  00           C            DEFB   0                ;GRAFIC MODE 2
1C57  00           C            DEFB   0                ;RESET VIDEO I (COLDSTART)
1C58  18           C            DEFB   CAN              ;ERASE TO  END  OF SCREEN
1C59  4C           C            DEFB   'L'              ;ERASE TO START OF SCREEN       ++
1C5A  00           C            DEFB   0                ;ERASE SCREEN
1C5B  1C           C            DEFB   FS               ;ERASE SCREEN + CURSOR HOME
1C5C  12           C            DEFB   DC2              ;CURSOR HOME
1C5D  0F           C            DEFB   SI               ;ERASE TO END OF LINE
1C5E  00           C            DEFB   0                ;ERASE LINE
1C5F  00 00        C            DEFB   0, 0             ;PAGE           UP/DOWN
1C61  00           C            DEFB   0                ;PAGE           ACTUAL
1C62  00 00        C            DEFB   0, 0             ;SCROLL         UP/DOWN
1C64  13 1A        C            DEFB   DC3, SUB         ;PARTIAL SCROLL UP/DOWN
1C66  05           C            DEFB   ENQ              ;SEND CURSOR ADDRESS WITHOUT OFFSET =) X ; Y
1C67  00           C            DEFB   0                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H =) X ; Y
1C68  00           C            DEFB   0                ;SEND CURSOR ADDRESS WITHOUT OFFSET =) Y ; X
1C69  00           C            DEFB   0                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H =) Y ; X
1C6A  21           C            DEFB   '!'              ;SEND CHAR AT CURSOR
1C6B  00           C            DEFB   0                ;CURSOR FORWARD
1C6C  00 00        C            DEFB   0, 0             ;CURSOR RIGHT/LEFT
1C6E  0C 0B        C            DEFB   FF, VT           ;CURSOR UP/DOWN
1C70  00 00        C            DEFB   0, 0             ;RESERVED: DO NOT USE
1C72  00           C            DEFB   0                ;VERTICAL   CURSOR ADDRESSING
1C73  00           C            DEFB   0                ;HORIZONTAL CURSOR ADDRESSING
1C74  11           C            DEFB   DC1              ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET =) X ; Y
1C75  00           C            DEFB   0                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H =) X ; Y
1C76  00           C            DEFB   0                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET =) Y ; X
1C77  00           C            DEFB   0                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H =) Y ; X
1C78  00           C            DEFB   0                ;CARRIAGE RETURN + ERASE END OF LINE
1C79  00           C            DEFB   0                ;CARRIAGE RETURN + LINE FEED
1C7A  0D           C            DEFB   CR               ;CARRIAGE RETURN
1C7B  00           C            DEFB   0                ;REVERSE LINEFEED
1C7C  00           C            DEFB   0                ;LINEFEED
1C7D  00           C            DEFB   0                ;BACKSPACE
1C7E  58           C            DEFB   'X'              ;TWO BYTE SEQUENCE              ++
1C7F  00           C    T2ESC:  DEFB   0                ;BYTE (ESCAPE) SEQUENCE
                   C            PAGE
```

```
                        C
                        C       ;-----------------------------------------------------------
                        C       ;       TERMINAL 2      TWO BYTE SEQUENCES, (ESC x) - SEQUENCES
                        C       ;                       NOT COMPATIBLE WITH VISA
                        C       ;-----------------------------------------------------------
                        C               REPT    128-ENTRIES
                        C               DEFB    0                       ;RESERVED
                        C               ENDM
1C80    00              C+              DEFB    0                       ;RESERVED
1C81    00              C+              DEFB    0                       ;RESERVED
1C82    00              C+              DEFB    0                       ;RESERVED
1C83    00              C+              DEFB    0                       ;RESERVED
1C84    00              C+              DEFB    0                       ;RESERVED
1C85    00              C+              DEFB    0                       ;RESERVED
1C86    00              C+              DEFB    0                       ;RESERVED
1C87    00              C+              DEFB    0                       ;RESERVED
1C88    00              C+              DEFB    0                       ;RESERVED
1C89    00              C+              DEFB    0                       ;RESERVED
1C8A    00              C+              DEFB    0                       ;RESERVED
1C8B    00              C+              DEFB    0                       ;RESERVED
1C8C    00              C+              DEFB    0                       ;RESERVED
1C8D    73              C               defb    115                     ;reset cursor stack
1C8E    72              C               defb    114                     ;swap cursor
1C8F    71 70           C               defb    113, 112                ;pop/push cursor
1C91    6F 6E 6D        C               DEFB    111,110,109             ;40/48/ 66 CHARACTERS PER LINE (12 LINES PER SCREEN)
1C94    6C 6B 6A        C               DEFB    108,107,106             ;80/96/132 CHARACTERS PER LINE (12 LINES PER SCREEN)
1C97    69 68 67        C               DEFB    105,104,103             ;40/48/ 66 CHARACTERS PER LINE (24 LINES PER SCREEN)
1C9A    66 65 64        C               DEFB    102,101,100             ;80/96/132 CHARACTERS PER LINE (25 LINES PER SCREEN)
1C9D    63 62           C               DEFB     99, 98                 ;SCREEN DARK    ON/OFF
1C9F    61              C               DEFB     97                     ;CLEAR WINDOW
1CA0    60 5F           C               DEFB     96, 95                 ;WINDOW SCROLL  UP/DOWN
1CA2    5E              C               defb     94                     ;load 6845
1CA3    5D              C               DEFB     93                     ;IGNORE NEXT CHAR
1CA4    5C              C               DEFB     92                     ;DISPLAY TEST PATTERN
1CA5    5B              C               DEFB     91                     ;CLEAR KEYBOARD BUFFER
1CA6    5A 59           C               DEFB     90, 89                 ;KEYBOARD BUFFER ACTIVE/NOT ACTIV
1CA8    58              C               DEFB     88                     ;RING BELL
1CA9    57              C               DEFB     87                     ;CURSOR ON
1CAA    56              C               DEFB     86                     ;CURSOR BLINK ON
1CAB    55              C               DEFB     85                     ;CURSOR OFF
1CAC    54 53           C               DEFB     84, 83                 ;KEYBOARD       UNLOCK/LOCK
1CAE    52 51           C               DEFB     82, 81                 ;INVERS SCREEN  ON/OFF
1CB0    50 4F           C               DEFB     80, 79                 ;CHAR INVERS    ON/OFF
1CB2    4E 4D           C               DEFB     78, 77                 ;BLINKING CHAR  ON/OFF
1CB4    4C 4B           C               DEFB     76, 75                 ;UNDERLINE      ON/OFF
1CB6    4A 49           C               DEFB     74, 73                 ;HALF INTENSITY ON/OFF
1CB8    48              C               DEFB     72                     ;NORMAL VIDEO
1CB9    47              C               DEFB     71                     ;CHAR INVERS ONLY
1CBA    46              C               DEFB     70                     ;BLINK ONLY
1CBB    45              C               DEFB     69                     ;UNDERLINE ONLY
1CBC    44              C               DEFB     68                     ;HALF INTENSITY ONLY
1CBD    43              C               DEFB     67                     ;HALF INTENSITY BLINK
1CBE    42              C               DEFB     66                     ;CHAR INVERS + HALF INTENSITY
1CBF    41              C               DEFB     65                     ;CHAR INVERS + BLINK
1CC0    40              C               DEFB     64                     ;CHAR INVERS + HALF INT. + BLINK
1CC1    3F              C               DEFB     63                     ;UNDERLINE + HALF INTENSITY
1CC2    3E              C               DEFB     62                     ;UNDERLINE BLINK
1CC3    3D              C               DEFB     61                     ;UNDERLINE + HALF INT. + BLINK
1CC4    3C              C               DEFB     60                     ;IDENTIFICATION
1CC5    3B              C               DEFB     59                     ;LOAD STRING
1CC6    3A              C               DEFB     58                     ;CLEAR STRING BUFFER
```

```
1CC7  39          C            DEFB   57              ;FORMFEED
1CC8  38          C            DEFB   56              ;CHANGE CHARACTER CODE SET (KEYBOARD)
1CC9  37          C            DEFB   55              ;CHANGE CHARACTER CODE SET (SCREEN)
1CCA  36 35       C            DEFB   54, 53          ;GERMAN/US CHARACTER SET
1CCC  34 33 32    C            DEFB   52, 51, 50      ;80/96/132 CHARACTERS PER LINE (24 LINES PER SCREEN)
1CCF  31          C            DEFB   49              ;ENABLE TERMINAL1
1CD0  30          C            DEFB   48              ;ENABLE TERMINAL2
1CD1  2F          C            DEFB   47              ;ENABLE TERMINAL3
1CD2  2E          C            DEFB   46              ;ENABLE TERMINAL4
1CD3  2D          C            DEFB   45              ;ENABLE TERMINAL5
1CD4  2C          C            DEFB   44              ;LOAD  TERMINAL5
1CD5  2B          C            DEFB   43              ;GRAFIC MODE 1
1CD6  2A          C            DEFB   42              ;GRAFIC MODE 2
1CD7  29          C            DEFB   41              ;RESET VIDEO I (COLDSTART)
1CD8  28          C            DEFB   40              ;ERASE TO  END  OF SCREEN
1CD9  27          C            DEFB   39              ;ERASE TO START OF SCREEN
1CDA  26          C            DEFB   38              ;ERASE SCREEN
1CDB  25          C            DEFB   37              ;ERASE SCREEN + CURSOR HOME
1CDC  24          C            DEFB   36              ;CURSOR HOME
1CDD  23          C            DEFB   35              ;ERASE TO END OF LINE
1CDE  22          C            DEFB   34              ;ERASE LINE
1CDF  21 20       C            DEFB   33, 32          ;PAGE          UP/DOWN
1CE1  1F          C            DEFB   31              ;PAGE          ACTUAL
1CE2  1E 1D       C            DEFB   30, 29          ;SCROLL        UP/DOWN
1CE4  1C 1B       C            DEFB   28, 27          ;PARTIAL SCROLL UP/DOWN
1CE6  1A          C            DEFB   26              ;SEND CURSOR ADDRESS WITHOUT OFFSET =) X ; Y
1CE7  19          C            DEFB   25              ;SEND CURSOR ADDRESS WITH OFFSET OF 20H =) X ; Y
1CE8  18          C            DEFB   24              ;SEND CURSOR ADDRESS WITHOUT OFFSET =) Y ; X
1CE9  17          C            DEFB   23              ;SEND CURSOR ADDRESS WITH OFFSET OF 20H =) Y ; X
1CEA  16          C            DEFB   22              ;SEND CHAR AT CURSOR
1CEB  15          C            DEFB   21              ;CURSOR FORWARD
1CEC  14 13       C            DEFB   20, 19          ;CURSOR RIGHT/LEFT
1CEE  12 11       C            DEFB   18, 17          ;CURSOR UP/DOWN
1CF0  10 0F       C            DEFB   16, 15          ;RESERVED: DO NOT USE
1CF2  0E          C            DEFB   14              ;VERTICAL   CURSOR ADDRESSING
1CF3  0D          C            DEFB   13              ;HORIZONTAL CURSOR ADDRESSING
1CF4  0C          C            DEFB   12              ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET =) X ; Y
1CF5  0B          C            DEFB   11              ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H =) X ; Y
1CF6  0A          C            DEFB   10              ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET =) Y ; X
1CF7  09          C            DEFB    9              ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H =) Y ; X
1CF8  08          C            DEFB    8              ;CARRIAGE RETURN + ERASE END OF LINE
1CF9  07          C            DEFB    7              ;CARRIAGE RETURN + LINE FEED
1CFA  06          C            DEFB    6              ;CARRIAGE RETURN
1CFB  05          C            DEFB    5              ;REVERSE LINEFEED
1CFC  04          C            DEFB    4              ;LINEFEED
1CFD  03          C            DEFB    3              ;BACKSPACE
1CFE  00          C            DEFB   0;2             ;TWO BYTE SEQUENCE
1CFF  00          C  T2ESCX:   DEFB   0;1             ;BYTE (ESCAPE) SEQUENCE
```

```
                        C       ;-----------------------------------------------------
                        C       ;       TERMINAL 3 (LIKE ADM 3A + PARTS OF info-s VIDEO 7)
                        C       ;-----------------------------------------------------
                        C
1D00                    C       TABLE3: REPT    128-ENTRIES
                        C               DEFB    0                       ;RESERVED
                        C               ENDM
1D00    00              C+              DEFB    0                       ;RESERVED
1D01    00              C+              DEFB    0                       ;RESERVED
1D02    00              C+              DEFB    0                       ;RESERVED
1D03    00              C+              DEFB    0                       ;RESERVED
1D04    00              C+              DEFB    0                       ;RESERVED
1D05    00              C+              DEFB    0                       ;RESERVED
1D06    00              C+              DEFB    0                       ;RESERVED
1D07    00              C+              DEFB    0                       ;RESERVED
1D08    00              C+              DEFB    0                       ;RESERVED
1D09    00              C+              DEFB    0                       ;RESERVED
1D0A    00              C+              DEFB    0                       ;RESERVED
1D0B    00              C+              DEFB    0                       ;RESERVED
1D0C    00              C+              DEFB    0                       ;RESERVED
1D0D    00              C               defb    0                       ;reset cursor stack
1D0E    00              C               defb    0                       ;swap cursor
1D0F    00 00           C               defb    0   , 0                 ;pop/push cursor
1D11    00 00 00        C               DEFB    0   , 0   , 0           ;40/48/ 66 CHARACTERS PER LINE (12 LINES PER SCREEN)
1D14    00 00 00        C               DEFB    0   , 0   , 0           ;80/96/132 CHARACTERS PER LINE (12 LINES PER SCREEN)
1D17    00 00 00        C               DEFB    0   , 0   , 0           ;40/48/ 66 CHARACTERS PER LINE (24 LINES PER SCREEN)
1D1A    00 00 00        C               DEFB    0   , 0   , 0           ;80/96/132 CHARACTERS PER LINE (25 LINES PER SCREEN)
1D1D    00 00           C               DEFB    0   , 0                 ;SCREEN DARK    ON/OFF
1D1F    00              C               DEFB    0                       ;CLEAR WINDOW
1D20    00 00           C               DEFB    0   , 0                 ;WINDOW SCROLL  UP/DOWN
1D22    00              C               defb    0                       ;load 6845
1D23    00              C               DEFB    0                       ;IGNORE NEXT CHAR
1D24    00              C               DEFB    0                       ;DISPLAY TEST PATTERN
1D25    00              C               DEFB    0                       ;CLEAR KEYBOARD BUFFER
1D26    00 00           C               DEFB    0   , 0                 ;KEYBOARD BUFFER ACTIVE/NOT ACTIV
1D28    07              C               DEFB    BEL                     ;RING BELL
1D29    00              C               DEFB    0                       ;CURSOR ON
1D2A    00              C               DEFB    0                       ;CURSOR BLINK ON
1D2B    00              C               DEFB    0                       ;CURSOR OFF
1D2C    00 00           C               DEFB    0   , 0                 ;KEYBOARD       UNLOCK/LOCK
1D2E    00 00           C               DEFB    0   , 0                 ;INVERS SCREEN  ON/OFF
1D30    00 00           C               DEFB    0   , 0                 ;CHAR INVERS    ON/OFF
1D32    00 00           C               DEFB    0   , 0                 ;BLINKING CHAR  ON/OFF
1D34    00 00           C               DEFB    0   , 0                 ;UNDERLINE      ON/OFF
1D36    00 00           C               DEFB    0   , 0                 ;HALF INTENSITY ON/OFF
1D38    00              C               DEFB    0                       ;NORMAL VIDEO
1D39    00              C               DEFB    0                       ;CHAR INVERS ONLY
1D3A    00              C               DEFB    0                       ;BLINK ONLY
1D3B    00              C               DEFB    0                       ;UNDERLINE ONLY
1D3C    00              C               DEFB    0                       ;HALF INTENSITY ONLY
1D3D    00              C               DEFB    0                       ;HALF INTENSITY BLINK
1D3E    00              C               DEFB    0                       ;CHAR INVERS + HALF INTENSITY
1D3F    00              C               DEFB    0                       ;CHAR INVERS + BLINK
1D40    00              C               DEFB    0                       ;CHAR INVERS + HALF INT. + BLINK
1D41    00              C               DEFB    0                       ;UNDERLINE + HALF INTENSITY
1D42    00              C               DEFB    0                       ;UNDERLINE BLINK
1D43    00              C               DEFB    0                       ;UNDERLINE + HALF INT. + BLINK
1D44    00              C               DEFB    0                       ;IDENTIFICATION
1D45    00              C               DEFB    0                       ;LOAD STRING
1D46    00              C               DEFB    0                       ;CLEAR STRING BUFFER
1D47    00              C               DEFB    0                       ;FORMFEED
1D48    00              C               DEFB    0                       ;CHANGE CHARACTER CODE SET (KEYBOARD)
```

```
1D49  00           C            DEFB  0                ;CHANGE CHARACTER CODE SET (SCREEN)
1D4A  00 00        C            DEFB  0   , 0           ;GERMAN/US CHARACTER SET
1D4C  00 00 00     C            DEFB  0   , 0   , 0     ;80/96/132 CHARACTERS PER LINE (24 LINES PER SCREEN)
1D4F  00           C            DEFB  0                ;ENABLE TERMINAL1
1D50  00           C            DEFB  0                ;ENABLE TERMINAL2
1D51  00           C            DEFB  0                ;ENABLE TERMINAL3
1D52  00           C            DEFB  0                ;ENABLE TERMINAL4
1D53  00           C            DEFB  0                ;ENABLE TERMINAL5
1D54  00           C            DEFB  0                ;LOAD  TERMINAL5
1D55  00           C            DEFB  0                ;GRAFIC MODE 1
1D56  00           C            DEFB  0                ;GRAFIC MODE 2
1D57  00           C            DEFB  0                ;RESET VIDEO I (COLDSTART)
1D58  18           C            DEFB  CAN              ;ERASE TO  END  OF SCREEN
1D59  00           C            DEFB  0                ;ERASE TO START OF SCREEN
1D5A  00           C            DEFB  0                ;ERASE SCREEN
1D5B  0C           C            DEFB  FF               ;ERASE SCREEN + CURSOR HOME
1D5C  1C           C            DEFB  FS               ;CURSOR HOME
1D5D  17           C            DEFB  ETB              ;ERASE TO END OF LINE
1D5E  1E           C            DEFB  RS               ;ERASE LINE
1D5F  00 00        C            DEFB  0   , 0          ;PAGE          UP/DOWN
1D61  00           C            DEFB  0                ;PAGE          ACTUAL
1D62  00 00        C            DEFB  0   , 0          ;SCROLL        UP/DOWN
1D64  1A 19        C            DEFB  SUB , EM         ;PARTIAL SCROLL UP/DOWN
1D66  00           C            DEFB  0                ;SEND CURSOR ADDRESS WITHOUT OFFSET ==) X ; Y
1D67  00           C            DEFB  0                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H ==) X ; Y
1D68  00           C            DEFB  0                ;SEND CURSOR ADDRESS WITHOUT OFFSET ==) Y ; X
1D69  00           C            DEFB  0                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H ==) Y ; X
1D6A  00           C            DEFB  0                ;SEND CHAR AT CURSOR
1D6B  09           C            DEFB  HT               ;CURSOR FORWARD
1D6C  00 00        C            DEFB  0   , 0          ;CURSOR RIGHT/LEFT
1D6E  0B 00        C            DEFB  VT  , 0          ;CURSOR UP/DOWN
1D70  00 00        C            DEFB  0   , 0          ;RESERVED: DO NOT USE
1D72  00           C            DEFB  0                ;VERTICAL   CURSOR ADDRESSING
1D73  00           C            DEFB  0                ;HORIZONTAL CURSOR ADDRESSING
1D74  00           C            DEFB  0                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET ==) X ; Y
1D75  00           C            DEFB  0                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H ==) X ; Y
1D76  00           C            DEFB  0                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET ==) Y ; X
1D77  00           C            DEFB  0                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H ==) Y ; X
1D78  00           C            DEFB  0                ;CARRIAGE RETURN + ERASE END OF LINE
1D79  00           C            DEFB  0                ;CARRIAGE RETURN + LINE FEED
1D7A  0D           C            DEFB  CR               ;CARRIAGE RETURN
1D7B  00           C            DEFB  0                ;REVERSE LINEFEED
1D7C  0A           C            DEFB  LF               ;LINEFEED
1D7D  08           C            DEFB  BS               ;BACKSPACE
1D7E  00           C            DEFB  0                ;TWO BYTE SEQUENCE
1D7F  1B           C    T3CON:  DEFB  ESC              ;BYTE (ESCAPE) SEQUENCE
                   C
                   C            PAGE
```

```
                        C
                        C       ;---------------------------------------------------------
                        C       ;       TERMINAL 3      BYTE (ESC) - SEQUENCES
                        C       ;---------------------------------------------------------
                        C
                        C               REPT    128-ENTRIES
                        C               DEFB    0                       ;RESERVED
                        C               ENDM
1D80    00              C+              DEFB    0                       ;RESERVED
1D81    00              C+              DEFB    0                       ;RESERVED
1D82    00              C+              DEFB    0                       ;RESERVED
1D83    00              C+              DEFB    0                       ;RESERVED
1D84    00              C+              DEFB    0                       ;RESERVED
1D85    00              C+              DEFB    0                       ;RESERVED
1D86    00              C+              DEFB    0                       ;RESERVED
1D87    00              C+              DEFB    0                       ;RESERVED
1D88    00              C+              DEFB    0                       ;RESERVED
1D89    00              C+              DEFB    0                       ;RESERVED
1D8A    00              C+              DEFB    0                       ;RESERVED
1D8B    00              C+              DEFB    0                       ;RESERVED
1D8C    00              C+              DEFB    0                       ;RESERVED
1D8D    00              C               defb    0                       ;reset cursor stack
1D8E    00              C               defb    0                       ;swap cursor
1D8F    6A 69           C               defb    'j' , 'i'               ;pop/push cursor
1D91    55 00 00        C               DEFB    'U' , 0   , 0           ;40/48/ 66 CHARACTERS PER LINE (12 LINES PER SCREEN)
1D94    00 00 00        C               DEFB    0   , 0   , 0           ;80/96/132 CHARACTERS PER LINE (12 LINES PER SCREEN)
1D97    54 00 00        C               DEFB    'T' , 0   , 0           ;40/48/ 66 CHARACTERS PER LINE (24 LINES PER SCREEN)
1D9A    00 00 00        C               DEFB    0   , 0   , 0           ;80/96/132 CHARACTERS PER LINE (25 LINES PER SCREEN)
1D9D    00 00           C               DEFB    0   , 0                 ;SCREEN DARK    ON/OFF
1D9F    00              C               DEFB    0                       ;CLEAR WINDOW
1DA0    00 00           C               DEFB    0   , 0                 ;WINDOW SCROLL  UP/DOWN
1DA2    00              C               defb    0                       ;load 6845
1DA3    00              C               DEFB    0                       ;IGNORE NEXT CHAR
1DA4    00              C               DEFB    0                       ;DISPLAY TEST PATTERN
1DA5    00              C               DEFB    0                       ;CLEAR KEYBOARD BUFFER
1DA6    00 00           C               DEFB    0   , 0                 ;KEYBOARD BUFFER ACTIVE/NOT ACTIV
1DA8    00              C               DEFB    0                       ;RING BELL
1DA9    4F              C               DEFB    'O'                     ;CURSOR ON
1DAA    4D              C               DEFB    'M'                     ;CURSOR BLINK ON
1DAB    50              C               DEFB    'P'                     ;CURSOR OFF
1DAC    67 79           C               DEFB    'g' , 'y'               ;KEYBOARD       UNLOCK/LOCK
1DAE    49 48           C               DEFB    'I' , 'H'               ;INVERS SCREEN  ON/OFF
1DB0    34 00           C               DEFB    '4' , 0                 ;CHAR INVERS    ON/OFF
1DB2    33 00           C               DEFB    '3' , 0                 ;BLINKING CHAR  ON/OFF
1DB4    35 00           C               DEFB    '5' , 0                 ;UNDERLINE      ON/OFF
1DB6    32 00           C               DEFB    '2' , 0                 ;HALF INTENSITY ON/OFF
1DB8    00              C               DEFB    0                       ;NORMAL VIDEO
1DB9    00              C               DEFB    0                       ;CHAR INVERS ONLY
1DBA    00              C               DEFB    0                       ;BLINK ONLY
1DBB    00              C               DEFB    0                       ;UNDERLINE ONLY
1DBC    00              C               DEFB    0                       ;HALF INTENSITY ONLY
1DBD    00              C               DEFB    0                       ;HALF INTENSITY BLINK
1DBE    00              C               DEFB    0                       ;CHAR INVERS + HALF INTENSITY
1DBF    00              C               DEFB    0                       ;CHAR INVERS + BLINK
1DC0    00              C               DEFB    0                       ;CHAR INVERS + HALF INT. + BLINK
1DC1    00              C               DEFB    0                       ;UNDERLINE + HALF INTENSITY
1DC2    00              C               DEFB    0                       ;UNDERLINE BLINK
1DC3    00              C               DEFB    0                       ;UNDERLINE + HALF INT. + BLINK
1DC4    00              C               DEFB    0                       ;IDENTIFICATION
1DC5    00              C               DEFB    0                       ;LOAD STRING
1DC6    70              C               DEFB    'p'                     ;CLEAR STRING BUFFER
```

```
1DC7  00           C            DEFB   0                ;FORMFEED
1DC8  00           C            DEFB   0                ;CHANGE CHARACTER CODE SET (KEYBOARD)
1DC9  00           C            DEFB   0                ;CHANGE CHARACTER CODE SET (SCREEN)
1DCA  59 58        C            DEFB   'Y' , 'X'        ;GERMAN/US CHARACTER SET
1DCC  53 00 00     C            DEFB   'S' , 0   , 0    ;80/96/132 CHARACTERS PER LINE (24 LINES PER SCREEN)
1DCF  00           C            DEFB   0                ;ENABLE TERMINAL1
1DD0  00           C            DEFB   0                ;ENABLE TERMINAL2
1DD1  00           C            DEFB   0                ;ENABLE TERMINAL3
1DD2  00           C            DEFB   0                ;ENABLE TERMINAL4
1DD3  00           C            DEFB   0                ;ENABLE TERMINAL5
1DD4  00           C            DEFB   0                ;LOAD   TERMINAL5
1DD5  00           C            DEFB   0                ;GRAFIC MODE 1
1DD6  00           C            DEFB   0                ;GRAFIC MODE 2
1DD7  00           C            DEFB   0                ;RESET VIDEO I (COLDSTART)
1DD8  00           C            DEFB   0                ;ERASE TO  END  OF SCREEN
1DD9  00           C            DEFB   0                ;ERASE TO START OF SCREEN
1DDA  00           C            DEFB   0                ;ERASE SCREEN
1DDB  00           C            DEFB   0                ;ERASE SCREEN + CURSOR HOME
1DDC  00           C            DEFB   0                ;CURSOR HOME
1DDD  00           C            DEFB   0                ;ERASE TO END OF LINE
1DDE  00           C            DEFB   0                ;ERASE LINE
1DDF  00 00        C            DEFB   0   , 0          ;PAGE           UP/DOWN
1DE1  00           C            DEFB   0                ;PAGE           ACTUAL
1DE2  00 00        C            DEFB   0   , 0          ;SCROLL         UP/DOWN
1DE4  00 00        C            DEFB   0   , 0          ;PARTIAL SCROLL UP/DOWN
1DE6  00           C            DEFB   0                ;SEND CURSOR ADDRESS WITHOUT OFFSET ==) X ; Y
1DE7  00           C            DEFB   0                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H ==) X ; Y
1DE8  00           C            DEFB   0                ;SEND CURSOR ADDRESS WITHOUT OFFSET ==) Y ; X
1DE9  43           C            DEFB   'C'              ;SEND CURSOR ADDRESS WITH OFFSET OF 20H ==) Y ; X
1DEA  61           C            DEFB   'a'              ;SEND CHAR AT CURSOR
1DEB  00           C            DEFB   0                ;CURSOR FORWARD
1DEC  00 00        C            DEFB   0   , 0          ;CURSOR RIGHT/LEFT
1DEE  00 00        C            DEFB   0   , 0          ;CURSOR UP/DOWN
1DF0  00 00        C            DEFB   0   , 0          ;RESERVED: DO NOT USE
1DF2  00           C            DEFB   0                ;VERTICAL   CURSOR ADDRESSING
1DF3  00           C            DEFB   0                ;HORIZONTAL CURSOR ADDRESSING
1DF4  00           C            DEFB   0                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET ==) X ; Y
1DF5  00           C            DEFB   0                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H ==) X ; Y
1DF6  00           C            DEFB   0                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET ==) Y ; X
1DF7  3D           C            DEFB   '='              ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H ==) Y ; X
1DF8  00           C            DEFB   0                ;CARRIAGE RETURN + ERASE END OF LINE
1DF9  00           C            DEFB   0                ;CARRIAGE RETURN + LINE FEED
1DFA  00           C            DEFB   0                ;CARRIAGE RETURN
1DFB  00           C            DEFB   0                ;REVERSE LINEFEED
1DFC  00           C            DEFB   0                ;LINEFEED
1DFD  00           C            DEFB   0                ;BACKSPACE
1DFE  3F           C            DEFB   '?'              ;TWO BYTE SEQUENCE
1DFF  21           C    T3ESC:  DEFB   '!'              ;BYTE (ESCAPE) SEQUENCE
                   C            PAGE
```

```
                        C
                        C       ;-----------------------------------------------------------
                        C       ;       TERMINAL 3      TWO BYTE SEQUENCES, (ESC x) - SEQUENCES
                        C       ;-----------------------------------------------------------
                        C
                        C               REPT    128-ENTRIES
                        C               DEFB    0                       ;RESERVED
                        C               ENDM
1E00    00              C+              DEFB    0                       ;RESERVED
1E01    00              C+              DEFB    0                       ;RESERVED
1E02    00              C+              DEFB    0                       ;RESERVED
1E03    00              C+              DEFB    0                       ;RESERVED
1E04    00              C+              DEFB    0                       ;RESERVED
1E05    00              C+              DEFB    0                       ;RESERVED
1E06    00              C+              DEFB    0                       ;RESERVED
1E07    00              C+              DEFB    0                       ;RESERVED
1E08    00              C+              DEFB    0                       ;RESERVED
1E09    00              C+              DEFB    0                       ;RESERVED
1E0A    00              C+              DEFB    0                       ;RESERVED
1E0B    00              C+              DEFB    0                       ;RESERVED
1E0C    00              C+              DEFB    0                       ;RESERVED
1E0D    00              C               defb    0                       ;reset cursor stack
1E0E    00              C               defb    0                       ;swap cursor
1E0F    00 00           C               defb    0   , 0                 ;pop/push cursor
1E11    00 00 00        C               DEFB    0   , 0   , 0           ;40/48/ 66 CHARACTERS PER LINE (12 LINES PER SCREEN)
1E14    00 00 00        C               DEFB    0   , 0   , 0           ;80/96/132 CHARACTERS PER LINE (12 LINES PER SCREEN)
1E17    00 00 00        C               DEFB    0   , 0   , 0           ;40/48/ 66 CHARACTERS PER LINE (24 LINES PER SCREEN)
1E1A    00 00 00        C               DEFB    0   , 0   , 0           ;80/96/132 CHARACTERS PER LINE (25 LINES PER SCREEN)
1E1D    00 00           C               DEFB    0   , 0                 ;SCREEN DARK    ON/OFF
1E1F    00              C               DEFB    0                       ;CLEAR WINDOW
1E20    00 00           C               DEFB    0   , 0                 ;WINDOW SCROLL  UP/DOWN
1E22    00              C               defb    0                       ;load 6845
1E23    00              C               DEFB    0                       ;IGNORE NEXT CHAR
1E24    00              C               DEFB    0                       ;DISPLAY TEST PATTERN
1E25    00              C               DEFB    0                       ;CLEAR KEYBOARD BUFFER
1E26    00 00           C               DEFB    0   , 0                 ;KEYBOARD BUFFER ACTIVE/NOT ACTIV
1E28    00              C               DEFB    0                       ;RING BELL
1E29    00              C               DEFB    0                       ;CURSOR ON
1E2A    00              C               DEFB    0                       ;CURSOR BLINK ON
1E2B    00              C               DEFB    0                       ;CURSOR OFF
1E2C    00 00           C               DEFB    0   , 0                 ;KEYBOARD       UNLOCK/LOCK
1E2E    00 00           C               DEFB    0   , 0                 ;INVERS SCREEN  ON/OFF
1E30    00 34           C               DEFB    0   , '4'               ;CHAR INVERS    ON/OFF
1E32    00 33           C               DEFB    0   , '3'               ;BLINKING CHAR  ON/OFF
1E34    00 35           C               DEFB    0   , '5'               ;UNDERLINE      ON/OFF
1E36    00 32           C               DEFB    0   , '2'               ;HALF INTENSITY ON/OFF
1E38    00              C               DEFB    0                       ;NORMAL VIDEO
1E39    00              C               DEFB    0                       ;CHAR INVERS ONLY
1E3A    00              C               DEFB    0                       ;BLINK ONLY
1E3B    00              C               DEFB    0                       ;UNDERLINE ONLY
1E3C    00              C               DEFB    0                       ;HALF INTENSITY ONLY
1E3D    00              C               DEFB    0                       ;HALF INTENSITY BLINK
1E3E    00              C               DEFB    0                       ;CHAR INVERS + HALF INTENSITY
1E3F    00              C               DEFB    0                       ;CHAR INVERS + BLINK
1E40    00              C               DEFB    0                       ;CHAR INVERS + HALF INT. + BLINK
1E41    00              C               DEFB    0                       ;UNDERLINE + HALF INTENSITY
1E42    00              C               DEFB    0                       ;UNDERLINE BLINK
1E43    00              C               DEFB    0                       ;UNDERLINE + HALF INT. + BLINK
1E44    00              C               DEFB    0                       ;IDENTIFICATION
1E45    00              C               DEFB    0                       ;LOAD STRING
1E46    00              C               DEFB    0                       ;CLEAR STRING BUFFER
```

```
1E47  00           C            DEFB  0                ;FORMFEED
1E48  00           C            DEFB  0                ;CHANGE CHARACTER CODE SET (KEYBOARD)
1E49  00           C            DEFB  0                ;CHANGE CHARACTER CODE SET (SCREEN)
1E4A  00 00        C            DEFB  0   , 0          ;GERMAN/US CHARACTER SET
1E4C  00 00 00     C            DEFB  0   , 0   , 0    ;80/96/132 CHARACTERS PER LINE (24 LINES PER SCREEN)
1E4F  31           C            DEFB  49               ;ENABLE TERMINAL1
1E50  30           C            DEFB  48               ;ENABLE TERMINAL2
1E51  2F           C            DEFB  47               ;ENABLE TERMINAL3
1E52  2E           C            DEFB  46               ;ENABLE TERMINAL4
1E53  2D           C            DEFB  45               ;ENABLE TERMINAL5
1E54  2C           C            DEFB  44               ;LOAD   TERMINAL5
1E55  00           C            DEFB  0                ;GRAFIC MODE 1
1E56  00           C            DEFB  0                ;GRAFIC MODE 2
1E57  00           C            DEFB  0                ;RESET VIDEO I (COLDSTART)
1E58  00           C            DEFB  0                ;ERASE TO  END  OF SCREEN
1E59  00           C            DEFB  0                ;ERASE TO START OF SCREEN
1E5A  00           C            DEFB  0                ;ERASE SCREEN
1E5B  00           C            DEFB  0                ;ERASE SCREEN + CURSOR HOME
1E5C  00           C            DEFB  0                ;CURSOR HOME
1E5D  00           C            DEFB  0                ;ERASE TO END OF LINE
1E5E  00           C            DEFB  0                ;ERASE LINE
1E5F  00 00        C            DEFB  0   , 0          ;PAGE           UP/DOWN
1E61  00           C            DEFB  0                ;PAGE           ACTUAL
1E62  00 00        C            DEFB  0   , 0          ;SCROLL         UP/DOWN
1E64  00 00        C            DEFB  0   , 0          ;PARTIAL SCROLL UP/DOWN
1E66  00           C            DEFB  0                ;SEND CURSOR ADDRESS WITHOUT OFFSET => X ; Y
1E67  00           C            DEFB  0                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H => X ; Y
1E68  00           C            DEFB  0                ;SEND CURSOR ADDRESS WITHOUT OFFSET => Y ; X
1E69  00           C            DEFB  0                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H => Y ; X
1E6A  00           C            DEFB  0                ;SEND CHAR AT CURSOR
1E6B  00           C            DEFB  0                ;CURSOR FORWARD
1E6C  00 00        C            DEFB  0   , 0          ;CURSOR RIGHT/LEFT
1E6E  00 00        C            DEFB  0   , 0          ;CURSOR UP/DOWN
1E70  00 00        C            DEFB  0   , 0          ;RESERVED: DO NOT USE
1E72  00           C            DEFB  0                ;VERTICAL   CURSOR ADDRESSING
1E73  00           C            DEFB  0                ;HORIZONTAL CURSOR ADDRESSING
1E74  00           C            DEFB  0                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET => X ; Y
1E75  00           C            DEFB  0                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H => X ; Y
1E76  00           C            DEFB  0                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET => Y ; X
1E77  00           C            DEFB  0                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H => Y ; X
1E78  00           C            DEFB  0                ;CARRIAGE RETURN + ERASE END OF LINE
1E79  00           C            DEFB  0                ;CARRIAGE RETURN + LINE FEED
1E7A  00           C            DEFB  0                ;CARRIAGE RETURN
1E7B  00           C            DEFB  0                ;REVERSE LINEFEED
1E7C  00           C            DEFB  0                ;LINEFEED
1E7D  00           C            DEFB  0                ;BACKSPACE
1E7E  00           C            DEFB  0                ;TWO BYTE SEQUENCE
1E7F  00           C  T3ESCX:   DEFB  0                ;BYTE (ESCAPE) SEQUENCE
```

```
                        C       ;-------------------------------------------------------
                        C       ;       TERMINAL 4 (NOT DEFINED -- FOR USER)
                        C       ;-------------------------------------------------------
                        C
1E80                    C       TABLE4: REPT    128-ENTRIES
                        C               DEFB    0FFH                    ;RESERVED
                        C               ENDM
1E80    FF              C+              DEFB    0FFH                    ;RESERVED
1E81    FF              C+              DEFB    0FFH                    ;RESERVED
1E82    FF              C+              DEFB    0FFH                    ;RESERVED
1E83    FF              C+              DEFB    0FFH                    ;RESERVED
1E84    FF              C+              DEFB    0FFH                    ;RESERVED
1E85    FF              C+              DEFB    0FFH                    ;RESERVED
1E86    FF              C+              DEFB    0FFH                    ;RESERVED
1E87    FF              C+              DEFB    0FFH                    ;RESERVED
1E88    FF              C+              DEFB    0FFH                    ;RESERVED
1E89    FF              C+              DEFB    0FFH                    ;RESERVED
1E8A    FF              C+              DEFB    0FFH                    ;RESERVED
1E8B    FF              C+              DEFB    0FFH                    ;RESERVED
1E8C    FF              C+              DEFB    0FFH                    ;RESERVED
1E8D    FF              C               defb    0FFH                    ;reset cursor stack
1E8E    FF              C               defb    0FFH                    ;swap cursor
1E8F    FF FF           C               defb    0FFH, 0FFH              ;pop/push cursor
1E91    FF FF FF        C               DEFB    0FFH, 0FFH, 0FFH        ;40/48/ 66 CHARACTERS PER LINE (12 LINES PER SCREEN)
1E94    FF FF FF        C               DEFB    0FFH, 0FFH, 0FFH        ;80/96/132 CHARACTERS PER LINE (12 LINES PER SCREEN)
1E97    FF FF FF        C               DEFB    0FFH, 0FFH, 0FFH        ;40/48/ 66 CHARACTERS PER LINE (24 LINES PER SCREEN)
1E9A    FF FF FF        C               DEFB    0FFH, 0FFH, 0FFH        ;80/96/132 CHARACTERS PER LINE (25 LINES PER SCREEN)
1E9D    FF FF           C               DEFB    0FFH, 0FFH              ;SCREEN DARK    ON/OFF
1E9F    FF              C               DEFB    0FFH                    ;CLEAR WINDOW
1EA0    FF FF           C               DEFB    0FFH, 0FFH              ;WINDOW SCROLL  UP/DOWN
1EA2    FF              C               defb    0FFH                    ;load 6845
1EA3    FF              C               DEFB    0FFH                    ;IGNORE NEXT CHAR
1EA4    FF              C               DEFB    0FFH                    ;DISPLAY TEST PATTERN
1EA5    FF              C               DEFB    0FFH                    ;CLEAR KEYBOARD BUFFER
1EA6    FF FF           C               DEFB    0FFH, 0FFH              ;KEYBOARD BUFFER ACTIVE/NOT ACTIV
1EA8    FF              C               DEFB    0FFH                    ;RING BELL
1EA9    FF              C               DEFB    0FFH                    ;CURSOR ON
1EAA    FF              C               DEFB    0FFH                    ;CURSOR BLINK ON
1EAB    FF              C               DEFB    0FFH                    ;CURSOR OFF
1EAC    FF FF           C               DEFB    0FFH, 0FFH              ;KEYBOARD       UNLOCK/LOCK
1EAE    FF FF           C               DEFB    0FFH ,0FFH              ;INVERS SCREEN  ON/OFF
1EB0    FF FF           C               DEFB    0FFH, 0FFH              ;CHAR INVERS    ON/OFF
1EB2    FF FF           C               DEFB    0FFH, 0FFH              ;BLINKING CHAR  ON/OFF
1EB4    FF FF           C               DEFB    0FFH, 0FFH              ;UNDERLINE      ON/OFF
1EB6    FF FF           C               DEFB    0FFH, 0FFH              ;HALF INTENSITY ON/OFF
1EB8    FF              C               DEFB    0FFH                    ;NORMAL VIDEO
1EB9    FF              C               DEFB    0FFH                    ;CHAR INVERS ONLY
1EBA    FF              C               DEFB    0FFH                    ;BLINK ONLY
1EBB    FF              C               DEFB    0FFH                    ;UNDERLINE ONLY
1EBC    FF              C               DEFB    0FFH                    ;HALF INTENSITY ONLY
1EBD    FF              C               DEFB    0FFH                    ;HALF INTENSITY BLINK
1EBE    FF              C               DEFB    0FFH                    ;CHAR INVERS + HALF INTENSITY
1EBF    FF              C               DEFB    0FFH                    ;CHAR INVERS + BLINK
1EC0    FF              C               DEFB    0FFH                    ;CHAR INVERS + HALF INT. + BLINK
1EC1    FF              C               DEFB    0FFH                    ;UNDERLINE + HALF INTENSITY
1EC2    FF              C               DEFB    0FFH                    ;UNDERLINE BLINK
1EC3    FF              C               DEFB    0FFH                    ;UNDERLINE + HALF INT. + BLINK
1EC4    FF              C               DEFB    0FFH                    ;IDENTIFICATION
1EC5    FF              C               DEFB    0FFH                    ;LOAD STRING
1EC6    FF              C               DEFB    0FFH                    ;CLEAR STRING BUFFER
1EC7    FF              C               DEFB    0FFH                    ;FORMFEED
1EC8    FF              C               DEFB    0FFH                    ;CHANGE CHARACTER CODE SET (KEYBOARD)
```

```
1EC9  FF           C            DEFB   0FFH                ;CHANGE CHARACTER CODE SET (SCREEN)
1ECA  FF FF        C            DEFB   0FFH, 0FFH          ;GERMAN/US CHARACTER SET
1ECC  FF FF FF     C            DEFB   0FFH, 0FFH, 0FFH    ;80/96/132 CHARACTERS PER LINE (24 LINES PER SCREEN)
1ECF  FF           C            DEFB   0FFH                ;ENABLE TERMINAL1
1ED0  FF           C            DEFB   0FFH                ;ENABLE TERMINAL2
1ED1  FF           C            DEFB   0FFH                ;ENABLE TERMINAL3
1ED2  FF           C            DEFB   0FFH                ;ENABLE TERMINAL4
1ED3  FF           C            DEFB   0FFH                ;ENABLE TERMINAL5
1ED4  FF           C            DEFB   0FFH                ;LOAD   TERMINAL5
1ED5  FF           C            DEFB   0FFH                ;GRAFIC MODE 1
1ED6  FF           C            DEFB   0FFH                ;GRAFIC MODE 2
1ED7  FF           C            DEFB   0FFH                ;RESET VIDEO I (COLDSTART)
1ED8  FF           C            DEFB   0FFH                ;ERASE TO  END  OF SCREEN
1ED9  FF           C            DEFB   0FFH                ;ERASE TO START OF SCREEN
1EDA  FF           C            DEFB   0FFH                ;ERASE SCREEN
1EDB  FF           C            DEFB   0FFH                ;ERASE SCREEN + CURSOR HOME
1EDC  FF           C            DEFB   0FFH                ;CURSOR HOME
1EDD  FF           C            DEFB   0FFH                ;ERASE TO END OF LINE
1EDE  FF           C            DEFB   0FFH                ;ERASE LINE
1EDF  FF FF        C            DEFB   0FFH, 0FFH          ;PAGE           UP/DOWN
1EE1  FF           C            DEFB   0FFH                ;PAGE           ACTUAL
1EE2  FF FF        C            DEFB   0FFH, 0FFH          ;SCROLL         UP/DOWN
1EE4  FF FF        C            DEFB   0FFH, 0FFH          ;PARTIAL SCROLL UP/DOWN
1EE6  FF           C            DEFB   0FFH                ;SEND CURSOR ADDRESS WITHOUT OFFSET => X ; Y
1EE7  FF           C            DEFB   0FFH                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H => X ; Y
1EE8  FF           C            DEFB   0FFH                ;SEND CURSOR ADDRESS WITHOUT OFFSET => Y ; X
1EE9  FF           C            DEFB   0FFH                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H => Y ; X
1EEA  FF           C            DEFB   0FFH                ;SEND CHAR AT CURSOR
1EEB  FF           C            DEFB   0FFH                ;CURSOR FORWARD
1EEC  FF FF        C            DEFB   0FFH, 0FFH          ;CURSOR RIGHT/LEFT
1EEE  FF FF        C            DEFB   0FFH, 0FFH          ;CURSOR UP/DOWN
1EF0  FF FF        C            DEFB   0FFH, 0FFH          ;RESERVED: DO NOT USE
1EF2  FF           C            DEFB   0FFH                ;VERTICAL   CURSOR ADDRESSING
1EF3  FF           C            DEFB   0FFH                ;HORIZONTAL CURSOR ADDRESSING
1EF4  FF           C            DEFB   0FFH                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET => X ; Y
1EF5  FF           C            DEFB   0FFH                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H => X ; Y
1EF6  FF           C            DEFB   0FFH                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET => Y ; X
1EF7  FF           C            DEFB   0FFH                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H => Y ; X
1EF8  FF           C            DEFB   0FFH                ;CARRIAGE RETURN + ERASE END OF LINE
1EF9  FF           C            DEFB   0FFH                ;CARRIAGE RETURN + LINE FEED
1EFA  FF           C            DEFB   0FFH                ;CARRIAGE RETURN
1EFB  FF           C            DEFB   0FFH                ;REVERSE LINEFEED
1EFC  FF           C            DEFB   0FFH                ;LINEFEED
1EFD  FF           C            DEFB   0FFH                ;BACKSPACE
1EFE  FF           C            DEFB   0FFH                ;TWO BYTE SEQUENCE
1EFF  FF           C  T4CON:    DEFB   0FFH                ;BYTE (ESCAPE) SEQUENCE
                   C            PAGE
```

```
                        C
                        C       ;-----------------------------------------------------
                        C       ;       TERMINAL 4      BYTE (ESC) - SEQUENCES
                        C       ;-----------------------------------------------------
                        C
                        C               REPT    128-ENTRIES
                        C               DEFB    0FFH                    ;RESERVED
                        C               ENDM
1F00    FF              C+              DEFB    0FFH                    ;RESERVED
1F01    FF              C+              DEFB    0FFH                    ;RESERVED
1F02    FF              C+              DEFB    0FFH                    ;RESERVED
1F03    FF              C+              DEFB    0FFH                    ;RESERVED
1F04    FF              C+              DEFB    0FFH                    ;RESERVED
1F05    FF              C+              DEFB    0FFH                    ;RESERVED
1F06    FF              C+              DEFB    0FFH                    ;RESERVED
1F07    FF              C+              DEFB    0FFH                    ;RESERVED
1F08    FF              C+              DEFB    0FFH                    ;RESERVED
1F09    FF              C+              DEFB    0FFH                    ;RESERVED
1F0A    FF              C+              DEFB    0FFH                    ;RESERVED
1F0B    FF              C+              DEFB    0FFH                    ;RESERVED
1F0C    FF              C+              DEFB    0FFH                    ;RESERVED
1F0D    FF              C               defb    0FFH                    ;reset cursor stack
1F0E    FF              C               defb    0FFH                    ;swap cursor
1F0F    FF FF           C               defb    0FFH, 0FFH              ;pop/push cursor
1F11    FF FF FF        C               DEFB    0FFH, 0FFH, 0FFH        ;40/48/ 66 CHARACTERS PER LINE (12 LINES PER SCREEN)
1F14    FF FF FF        C               DEFB    0FFH, 0FFH, 0FFH        ;80/96/132 CHARACTERS PER LINE (12 LINES PER SCREEN)
1F17    FF FF FF        C               DEFB    0FFH, 0FFH, 0FFH        ;40/48/ 66 CHARACTERS PER LINE (24 LINES PER SCREEN)
1F1A    FF FF FF        C               DEFB    0FFH, 0FFH, 0FFH        ;80/96/132 CHARACTERS PER LINE (25 LINES PER SCREEN)
1F1D    FF FF           C               DEFB    0FFH, 0FFH              ;SCREEN DARK    ON/OFF
1F1F    FF              C               DEFB    0FFH                    ;CLEAR WINDOW
1F20    FF FF           C               DEFB    0FFH, 0FFH              ;WINDOW SCROLL  UP/DOWN
1F22    FF              C               defb    0FFH                    ;load 6845
1F23    FF              C               DEFB    0FFH                    ;IGNORE NEXT CHAR
1F24    FF              C               DEFB    0FFH                    ;DISPLAY TEST PATTERN
1F25    FF              C               DEFB    0FFH                    ;CLEAR KEYBOARD BUFFER
1F26    FF FF           C               DEFB    0FFH, 0FFH              ;KEYBOARD BUFFER ACTIVE/NOT ACTIV
1F28    FF              C               DEFB    0FFH                    ;RING BELL
1F29    FF              C               DEFB    0FFH                    ;CURSOR ON
1F2A    FF              C               DEFB    0FFH                    ;CURSOR BLINK ON
1F2B    FF              C               DEFB    0FFH                    ;CURSOR OFF
1F2C    FF FF           C               DEFB    0FFH, 0FFH              ;KEYBOARD       UNLOCK/LOCK
1F2E    FF FF           C               DEFB    0FFH, 0FFH              ;INVERS SCREEN  ON/OFF
1F30    FF FF           C               DEFB    0FFH, 0FFH              ;CHAR INVERS    ON/OFF
1F32    FF FF           C               DEFB    0FFH, 0FFH              ;BLINKING CHAR  ON/OFF
1F34    FF FF           C               DEFB    0FFH, 0FFH              ;UNDERLINE      ON/OFF
1F36    FF FF           C               DEFB    0FFH, 0FFH              ;HALF INTENSITY ON/OFF
1F38    FF              C               DEFB    0FFH                    ;NORMAL VIDEO
1F39    FF              C               DEFB    0FFH                    ;CHAR INVERS ONLY
1F3A    FF              C               DEFB    0FFH                    ;BLINK ONLY
1F3B    FF              C               DEFB    0FFH                    ;UNDERLINE ONLY
1F3C    FF              C               DEFB    0FFH                    ;HALF INTENSITY ONLY
1F3D    FF              C               DEFB    0FFH                    ;HALF INTENSITY BLINK
1F3E    FF              C               DEFB    0FFH                    ;CHAR INVERS + HALF INTENSITY
1F3F    FF              C               DEFB    0FFH                    ;CHAR INVERS + BLINK
1F40    FF              C               DEFB    0FFH                    ;CHAR INVERS + HALF INT. + BLINK
1F41    FF              C               DEFB    0FFH                    ;UNDERLINE + HALF INTENSITY
1F42    FF              C               DEFB    0FFH                    ;UNDERLINE BLINK
1F43    FF              C               DEFB    0FFH                    ;UNDERLINE + HALF INT. + BLINK
1F44    FF              C               DEFB    0FFH                    ;IDENTIFICATION
1F45    FF              C               DEFB    0FFH                    ;LOAD STRING
1F46    FF              C               DEFB    0FFH                    ;CLEAR STRING BUFFER
```

```
1F47  FF           C            DEFB   0FFH                ;FORMFEED
1F48  FF           C            DEFB   0FFH                ;CHANGE CHARACTER CODE SET (KEYBOARD)
1F49  FF           C            DEFB   0FFH                ;CHANGE CHARACTER CODE SET (SCREEN)
1F4A  FF FF        C            DEFB   0FFH, 0FFH          ;GERMAN/US CHARACTER SET
1F4C  FF FF FF     C            DEFB   0FFH, 0FFH, 0FFH    ;80/96/132 CHARACTERS PER LINE (24 LINES PER SCREEN)
1F4F  FF           C            DEFB   0FFH                ;ENABLE TERMINAL1
1F50  FF           C            DEFB   0FFH                ;ENABLE TERMINAL2
1F51  FF           C            DEFB   0FFH                ;ENABLE TERMINAL3
1F52  FF           C            DEFB   0FFH                ;ENABLE TERMINAL4
1F53  FF           C            DEFB   0FFH                ;ENABLE TERMINAL5
1F54  FF           C            DEFB   0FFH                ;LOAD   TERMINAL5
1F55  FF           C            DEFB   0FFH                ;GRAFIC MODE 1
1F56  FF           C            DEFB   0FFH                ;GRAFIC MODE 2
1F57  FF           C            DEFB   0FFH                ;RESET VIDEO I (COLDSTART)
1F58  FF           C            DEFB   0FFH                ;ERASE TO  END  OF SCREEN
1F59  FF           C            DEFB   0FFH                ;ERASE TO START OF SCREEN
1F5A  FF           C            DEFB   0FFH                ;ERASE SCREEN
1F5B  FF           C            DEFB   0FFH                ;ERASE SCREEN + CURSOR HOME
1F5C  FF           C            DEFB   0FFH                ;CURSOR HOME
1F5D  FF           C            DEFB   0FFH                ;ERASE TO END OF LINE
1F5E  FF           C            DEFB   0FFH                ;ERASE LINE
1F5F  FF FF        C            DEFB   0FFH, 0FFH          ;PAGE          UP/DOWN
1F61  FF           C            DEFB   0FFH                ;PAGE          ACTUAL
1F62  FF FF        C            DEFB   0FFH, 0FFH          ;SCROLL        UP/DOWN
1F64  FF FF        C            DEFB   0FFH, 0FFH          ;PARTIAL SCROLL UP/DOWN
1F66  FF           C            DEFB   0FFH                ;SEND CURSOR ADDRESS WITHOUT OFFSET ==) X ; Y
1F67  FF           C            DEFB   0FFH                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H ==) X ; Y
1F68  FF           C            DEFB   0FFH                ;SEND CURSOR ADDRESS WITHOUT OFFSET ==) Y ; X
1F69  FF           C            DEFB   0FFH                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H ==) Y ; X
1F6A  FF           C            DEFB   0FFH                ;SEND CHAR AT CURSOR
1F6B  FF           C            DEFB   0FFH                ;CURSOR FORWARD
1F6C  FF FF        C            DEFB   0FFH, 0FFH          ;CURSOR RIGHT/LEFT
1F6E  FF FF        C            DEFB   0FFH, 0FFH          ;CURSOR UP/DOWN
1F70  FF FF        C            DEFB   0FFH, 0FFH          ;RESERVED: DO NOT USE
1F72  FF           C            DEFB   0FFH                ;VERTICAL   CURSOR ADDRESSING
1F73  FF           C            DEFB   0FFH                ;HORIZONTAL CURSOR ADDRESSING
1F74  FF           C            DEFB   0FFH                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET ==) X ; Y
1F75  FF           C            DEFB   0FFH                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H ==) X ; Y
1F76  FF           C            DEFB   0FFH                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET ==) Y ; X
1F77  FF           C            DEFB   0FFH                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H ==) Y ; X
1F78  FF           C            DEFB   0FFH                ;CARRIAGE RETURN + ERASE END OF LINE
1F79  FF           C            DEFB   0FFH                ;CARRIAGE RETURN + LINE FEED
1F7A  FF           C            DEFB   0FFH                ;CARRIAGE RETURN
1F7B  FF           C            DEFB   0FFH                ;REVERSE LINEFEED
1F7C  FF           C            DEFB   0FFH                ;LINEFEED
1F7D  FF           C            DEFB   0FFH                ;BACKSPACE
1F7E  FF           C            DEFB   0FFH                ;TWO BYTE SEQUENCE
1F7F  FF           C     T4ESC: DEFB   0FFH                ;BYTE (ESCAPE) SEQUENCE
                   C            PAGE
```

```
                        C
                        C       ;-----------------------------------------------------------
                        C       ;       TERMINAL 4      TWO BYTE SEQUENCES, (ESC x) - SEQUENCES
                        C       ;-----------------------------------------------------------
                        C
                        C               REPT    128-ENTRIES
                        C               DEFB    0FFH                    ;RESERVED
                        C               ENDM
1F80    FF              C+              DEFB    0FFH                    ;RESERVED
1F81    FF              C+              DEFB    0FFH                    ;RESERVED
1F82    FF              C+              DEFB    0FFH                    ;RESERVED
1F83    FF              C+              DEFB    0FFH                    ;RESERVED
1F84    FF              C+              DEFB    0FFH                    ;RESERVED
1F85    FF              C+              DEFB    0FFH                    ;RESERVED
1F86    FF              C+              DEFB    0FFH                    ;RESERVED
1F87    FF              C+              DEFB    0FFH                    ;RESERVED
1F88    FF              C+              DEFB    0FFH                    ;RESERVED
1F89    FF              C+              DEFB    0FFH                    ;RESERVED
1F8A    FF              C+              DEFB    0FFH                    ;RESERVED
1F8B    FF              C+              DEFB    0FFH                    ;RESERVED
1F8C    FF              C+              DEFB    0FFH                    ;RESERVED
1F8D    FF              C               defb    0FFH                    ;reset cursor stack
1F8E    FF              C               defb    0FFH                    ;swap cursor
1F8F    FF FF           C               defb    0FFH, 0FFH              ;pop/push cursor
1F91    FF FF FF        C               DEFB    0FFH, 0FFH, 0FFH        ;40/48/ 66 CHARACTERS PER LINE (12 LINES PER SCREEN)
1F94    FF FF FF        C               DEFB    0FFH, 0FFH, 0FFH        ;80/96/132 CHARACTERS PER LINE (12 LINES PER SCREEN)
1F97    FF FF FF        C               DEFB    0FFH, 0FFH, 0FFH        ;40/48/ 66 CHARACTERS PER LINE (24 LINES PER SCREEN)
1F9A    FF FF FF        C               DEFB    0FFH, 0FFH, 0FFH        ;80/96/132 CHARACTERS PER LINE (25 LINES PER SCREEN)
1F9D    FF FF           C               DEFB    0FFH, 0FFH              ;SCREEN DARK    ON/OFF
1F9F    FF              C               DEFB    0FFH                    ;CLEAR WINDOW
1FA0    FF FF           C               DEFB    0FFH, 0FFH              ;WINDOW SCROLL  UP/DOWN
1FA2    FF              C               defb    0FFH                    ;load 6845
1FA3    FF              C               DEFB    0FFH                    ;IGNORE NEXT CHAR
1FA4    FF              C               DEFB    0FFH                    ;DISPLAY TEST PATTERN
1FA5    FF              C               DEFB    0FFH                    ;CLEAR KEYBOARD BUFFER
1FA6    FF FF           C               DEFB    0FFH, 0FFH              ;KEYBOARD BUFFER ACTIVE/NOT ACTIV
1FA8    FF              C               DEFB    0FFH                    ;RING BELL
1FA9    FF              C               DEFB    0FFH                    ;CURSOR ON
1FAA    FF              C               DEFB    0FFH                    ;CURSOR BLINK ON
1FAB    FF              C               DEFB    0FFH                    ;CURSOR OFF
1FAC    FF FF           C               DEFB    0FFH, 0FFH              ;KEYBOARD       UNLOCK/LOCK
1FAE    FF FF           C               DEFB    0FFH ,0FFH              ;INVERS SCREEN  ON/OFF
1FB0    FF FF           C               DEFB    0FFH, 0FFH              ;CHAR INVERS    ON/OFF
1FB2    FF FF           C               DEFB    0FFH, 0FFH              ;BLINKING CHAR  ON/OFF
1FB4    FF FF           C               DEFB    0FFH, 0FFH              ;UNDERLINE      ON/OFF
1FB6    FF FF           C               DEFB    0FFH, 0FFH              ;HALF INTENSITY ON/OFF
1FB8    FF              C               DEFB    0FFH                    ;NORMAL VIDEO
1FB9    FF              C               DEFB    0FFH                    ;CHAR INVERS ONLY
1FBA    FF              C               DEFB    0FFH                    ;BLINK ONLY
1FBB    FF              C               DEFB    0FFH                    ;UNDERLINE ONLY
1FBC    FF              C               DEFB    0FFH                    ;HALF INTENSITY ONLY
1FBD    FF              C               DEFB    0FFH                    ;HALF INTENSITY BLINK
1FBE    FF              C               DEFB    0FFH                    ;CHAR INVERS + HALF INTENSITY
1FBF    FF              C               DEFB    0FFH                    ;CHAR INVERS + BLINK
1FC0    FF              C               DEFB    0FFH                    ;CHAR INVERS + HALF INT. + BLINK
1FC1    FF              C               DEFB    0FFH                    ;UNDERLINE + HALF INTENSITY
1FC2    FF              C               DEFB    0FFH                    ;UNDERLINE BLINK
1FC3    FF              C               DEFB    0FFH                    ;UNDERLINE + HALF INT. + BLINK
1FC4    FF              C               DEFB    0FFH                    ;IDENTIFICATION
1FC5    FF              C               DEFB    0FFH                    ;LOAD STRING
1FC6    FF              C               DEFB    0FFH                    ;CLEAR STRING BUFFER
```

```
1FC7  FF          C          DEFB   0FFH                ;FORMFEED
1FC8  FF          C          DEFB   0FFH                ;CHANGE CHARACTER CODE SET (KEYBOARD)
1FC9  FF          C          DEFB   0FFH                ;CHANGE CHARACTER CODE SET (SCREEN)
1FCA  FF FF       C          DEFB   0FFH, 0FFH          ;GERMAN/US CHARACTER SET
1FCC  FF FF FF    C          DEFB   0FFH, 0FFH, 0FFH    ;80/96/132 CHARACTERS PER LINE (24 LINES PER SCREEN)
1FCF  FF          C          DEFB   0FFH                ;ENABLE TERMINAL1
1FD0  FF          C          DEFB   0FFH                ;ENABLE TERMINAL2
1FD1  FF          C          DEFB   0FFH                ;ENABLE TERMINAL3
1FD2  FF          C          DEFB   0FFH                ;ENABLE TERMINAL4
1FD3  FF          C          DEFB   0FFH                ;ENABLE TERMINAL5
1FD4  FF          C          DEFB   0FFH                ;LOAD   TERMINAL5
1FD5  FF          C          DEFB   0FFH                ;GRAFIC MODE 1
1FD6  FF          C          DEFB   0FFH                ;GRAFIC MODE 2
1FD7  FF          C          DEFB   0FFH                ;RESET VIDEO I (COLDSTART)
1FD8  FF          C          DEFB   0FFH                ;ERASE TO  END  OF SCREEN
1FD9  FF          C          DEFB   0FFH                ;ERASE TO START OF SCREEN
1FDA  FF          C          DEFB   0FFH                ;ERASE SCREEN
1FDB  FF          C          DEFB   0FFH                ;ERASE SCREEN + CURSOR HOME
1FDC  FF          C          DEFB   0FFH                ;CURSOR HOME
1FDD  FF          C          DEFB   0FFH                ;ERASE TO END OF LINE
1FDE  FF          C          DEFB   0FFH                ;ERASE LINE
1FDF  FF FF       C          DEFB   0FFH, 0FFH          ;PAGE           UP/DOWN
1FE1  FF          C          DEFB   0FFH                ;PAGE           ACTUAL
1FE2  FF FF       C          DEFB   0FFH, 0FFH          ;SCROLL         UP/DOWN
1FE4  FF FF       C          DEFB   0FFH, 0FFH          ;PARTIAL SCROLL UP/DOWN
1FE6  FF          C          DEFB   0FFH                ;SEND CURSOR ADDRESS WITHOUT OFFSET => X ; Y
1FE7  FF          C          DEFB   0FFH                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H => X ; Y
1FE8  FF          C          DEFB   0FFH                ;SEND CURSOR ADDRESS WITHOUT OFFSET => Y ; X
1FE9  FF          C          DEFB   0FFH                ;SEND CURSOR ADDRESS WITH OFFSET OF 20H => Y ; X
1FEA  FF          C          DEFB   0FFH                ;SEND CHAR AT CURSOR
1FEB  FF          C          DEFB   0FFH                ;CURSOR FORWARD
1FEC  FF FF       C          DEFB   0FFH, 0FFH          ;CURSOR RIGHT/LEFT
1FEE  FF FF       C          DEFB   0FFH, 0FFH          ;CURSOR UP/DOWN
1FF0  FF FF       C          DEFB   0FFH, 0FFH          ;RESERVED: DO NOT USE
1FF2  FF          C          DEFB   0FFH                ;VERTICAL   CURSOR ADDRESSING
1FF3  FF          C          DEFB   0FFH                ;HORIZONTAL CURSOR ADDRESSING
1FF4  FF          C          DEFB   0FFH                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET => X ; Y
1FF5  FF          C          DEFB   0FFH                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H => X ; Y
1FF6  FF          C          DEFB   0FFH                ;ABSOLUTE   CURSOR ADDRESSING WITHOUT OFFSET => Y ; X
1FF7  FF          C          DEFB   0FFH                ;ABSOLUTE   CURSOR ADDRESSING WITH OFFSET 20H => Y ; X
1FF8  FF          C          DEFB   0FFH                ;CARRIAGE RETURN + ERASE END OF LINE
1FF9  FF          C          DEFB   0FFH                ;CARRIAGE RETURN + LINE FEED
1FFA  FF          C          DEFB   0FFH                ;CARRIAGE RETURN
1FFB  FF          C          DEFB   0FFH                ;REVERSE LINEFEED
1FFC  FF          C          DEFB   0FFH                ;LINEFEED
1FFD  FF          C          DEFB   0FFH                ;BACKSPACE
1FFE  FF          C          DEFB   0FFH                ;TWO BYTE SEQUENCE
1FFF  FF          C  T4ESCX: DEFB   0FFH                ;BYTE (ESCAPE) SEQUENCE
1FFF              LASTADR EQU    $-1                   ;M80 SHOULD PRINT 1FFF
                          END
```

Anhang F:

Tabellen der implementierten Zeichensätze

Z96 enthält die beiden Zeichensätze für 80/96 Zeichen pro Zeile und Z132 die beiden Zeichensätze für 132 Zeichen pro Zeile. Von 00H bis 7FH ist der jeweils 1. Zeichensatz und von 80H bis 0FFH der jeweils 2. Zeichensatz aufgelisted worden. Die angegebenen Codes entsprechen der internen Darstellung in der VIDEO I. Der Anwender kann vom Computer aus diese Zeichen jedoch in vielen Fällen nicht direkt mit der internen Codierung anwählen! So kann z.B. der jeweils 2. Zeichensatz nicht durch Setzen eines 8. Bites angewählt werden. Die Anwahl des 2. Zeichensatzes muß durch Auslösen einer geeigneten Funktion (z.B. Funktion 76) erfolgen.

CHARACTER GENERATION PROGRAM FOR JK82 VIDEO I VS 1.4 SET: Z96 V.2 25.04.85 16:37:59

ASCII VALUE = 00H

```
  0 1 2 3 4 5 6 7
0
1
2
3
4
5
6
7
8
9
A
```

ASCII VALUE = 01H

```
  0 1 2 3 4 5 6 7
0
1      XX
2    XXXXXX
3  XX  XX  XX
4      XX
5      XX
6      XX
7      XX
8      XX
9      XX
A      XX
```

ASCII VALUE = 02H

```
  0 1 2 3 4 5 6 7
0      XX
1      XX
2      XX
3      XX
4      XX
5      XX
6      XX
7  XX  XX  XX
8    XXXXXX
9      XX
A
```

ASCII VALUE = 03H

```
  0 1 2 3 4 5 6 7
0
1
2
3      XX
4    XX
5  XXXXXXXXXXXXXX
6    XX
7      XX
8
9
A
```

ASCII VALUE = 04H

```
  0 1 2 3 4 5 6 7
0
1
2
3         XX
4           XX
5XXXXXXXXXXXXXXX
6           XX
7         XX
8
9
A
```

ASCII VALUE = 05H

```
  0 1 2 3 4 5 6 7
0
1  XXXXXX
2  XXXX
3  XX  XX
4         XX
5           XX
6             XX
7               XX
8
9
A
```

ASCII VALUE = 06H

```
  0 1 2 3 4 5 6 7
0
1        XXXXXX
2          XXXX
3        XX  XX
4      XX
5    XX
6  XX
7XX
8
9
A
```

ASCII VALUE = 07H

```
  0 1 2 3 4 5 6 7
0
1XX
2  XX
3    XX
4      XX
5        XX  XX
6          XXXX
7        XXXXXX
8
9
A
```

ASCII VALUE = 08H

```
  0 1 2 3 4 5 6 7
0
1               XX
2             XX
3           XX
4         XX
5  XX  XX
6  XXXX
7  XXXXXX
8
9
A
```

ASCII VALUE = 09H

```
  0 1 2 3 4 5 6 7
0      XXXXXX
1    XX
2      XXXX
3    XX    XX
4      XX    XX
5        XXXX
6            XX
7      XXXXXX
8
9
A
```

ASCII VALUE = 0AH

```
  0 1 2 3 4 5 6 7
0
1
2
3    XX      XX
4    XX      XX
5    XX      XX
6    XX      XX
7    XXXXXXXX  XX
8    XX
9  XX
A
```

ASCII VALUE = 0BH

```
  0 1 2 3 4 5 6 7
0  XX        XX
1      XXXX
2    XX    XX
3  XX        XX
4  XX        XX
5  XXXXXXXXXXXX
6  XX        XX
7  XX        XX
8
9
A
```

ASCII VALUE = 0CH

```
  0 1 2 3 4 5 6 7
0  XX        XX
1
2    XXXXXXXX
3  XX        XX
4  XX        XX
5  XX        XX
6  XX        XX
7    XXXXXXXX
8
9
A
```

ASCII VALUE = 0DH

```
  0 1 2 3 4 5 6 7
0  XX        XX
1
2  XX        XX
3  XX        XX
4  XX        XX
5  XX        XX
6  XX        XX
7    XXXXXXXX
8
9
A
```

ASCII VALUE = 0EH

```
  0 1 2 3 4 5 6 7
0        XXXX
1      XX    XX
2      XX    XX
3        XXXX
4
5
6
7
8
9
A
```

ASCII VALUE = 0FH

```
  0 1 2 3 4 5 6 7
0      XXXX
1    XX    XX
2          XX
3      XXXX
4    XX
5    XXXXXXXX
6
7
8
9
A
```

ASCII VALUE = 10H

```
  0 1 2 3 4 5 6 7
0
1
2
3
4
5XXXXXXXXXXXXXXXX
6
7
8
9
A
```

ASCII VALUE = 11H

```
  0 1 2 3 4 5 6 7
0      XX
1      XX
2      XX
3      XX
4      XX
5      XX
6      XX
7      XX
8      XX
9      XX
A      XX
```

ASCII VALUE = 12H

```
  0 1 2 3 4 5 6 7
0      XX
1      XX
2      XX
3      XX
4      XX
5XXXXXXXXXXXXXXXX
6      XX
7      XX
8      XX
9      XX
A      XX
```

ASCII VALUE = 13H

```
  0 1 2 3 4 5 6 7
0      XX
1      XX
2      XX
3      XX
4      XX
5      XXXXXXXXXX
6
7
8
9
A
```

ASCII VALUE = 14H

```
  0 1 2 3 4 5 6 7
0
1
2
3
4
5XXXXXXXX
6      XX
7      XX
8      XX
9      XX
A      XX
```

ASCII VALUE = 15H

```
  0 1 2 3 4 5 6 7
0
1
2
3
4
5      XXXXXXXXXX
6      XX
7      XX
8      XX
9      XX
A      XX
```

ASCII VALUE = 16H

```
  0 1 2 3 4 5 6 7
0      XX
1      XX
2      XX
3      XX
4      XX
5XXXXXXXX
6
7
8
9
A
```

ASCII VALUE = 17H

```
  0 1 2 3 4 5 6 7
0      XX
1      XX
2      XX
3      XX
4      XX
5      XXXXXXXXXX
6      XX
7      XX
8      XX
9      XX
A      XX
```

ASCII VALUE = 18H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5XXXXXXXXXXXXXXXX
6      XX
7      XX
8      XX
9      XX
A      XX
```

ASCII VALUE = 19H

```
 0 1 2 3 4 5 6 7
0      XX
1      XX
2      XX
3      XX
4      XX
5XXXXXXXX
6      XX
7      XX
8      XX
9      XX
A      XX
```

ASCII VALUE = 1AH

```
 0 1 2 3 4 5 6 7
0      XX
1      XX
2      XX
3      XX
4      XX
5XXXXXXXXXXXXXXXX
6
7
8
9
A
```

ASCII VALUE = 1BH

```
 0 1 2 3 4 5 6 7
0
1    XX      XX
2
3      XXXXXXXX
4    XX      XX
5    XX      XX
6    XX      XX
7      XXXXXXXXXX
8
9
A
```

ASCII VALUE = 1CH

```
 0 1 2 3 4 5 6 7
0
1    XX      XX
2
3      XXXXXX
4    XX      XX
5    XX      XX
6    XX      XX
7      XXXXXX
8
9
A
```

ASCII VALUE = 1DH

```
 0 1 2 3 4 5 6 7
0
1    XX      XX
2
3    XX      XX
4    XX      XX
5    XX      XX
6    XX      XX
7      XXXXXXXX
8
9
A
```

ASCII VALUE = 1EH

```
 0 1 2 3 4 5 6 7
0
1      XXXXXX
2    XX      XX
3    XX      XX
4    XX  XXXX
5    XX      XX
6    XX      XX
7    XX  XXXX
8    XX
9
A
```

ASCII VALUE = 1FH

```
 0 1 2 3 4 5 6 7
0      XXXX
1    XX    XX
2        XX
3          XX
4    XX    XX
5      XXXX
6
7
8
9
A
```

ASCII VALUE = 20H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6
7
8
9
A
```

ASCII VALUE = 21H

```
 0 1 2 3 4 5 6 7
0
1        XX
2        XX
3        XX
4        XX
5        XX
6
7        XX
8
9
A
```

ASCII VALUE = 22H

```
 0 1 2 3 4 5 6 7
0
1      XX  XX
2      XX  XX
3
4
5
6
7
8
9
A
```

ASCII VALUE = 23H

```
 0 1 2 3 4 5 6 7
0
1      XX  XX
2      XX  XX
3    XXXXXXXXXX
4      XX  XX
5    XXXXXXXXXX
6      XX  XX
7      XX  XX
8
9
A
```

ASCII VALUE = 24H

```
 0 1 2 3 4 5 6 7
0
1        XX
2      XXXXXXXX
3    XX  XX
4      XXXXXX
5        XX  XX
6    XXXXXXXX
7        XX
8
9
A
```

ASCII VALUE = 25H

```
 0 1 2 3 4 5 6 7
0
1    XXXX      XX
2    XXXX    XX
3          XX
4        XX
5      XX
6    XX    XXXX
7  XX      XXXX
8
9
A
```

ASCII VALUE = 26H

```
 0 1 2 3 4 5 6 7
0
1      XX
2    XX  XX
3    XX  XX
4      XX
5    XX  XX  XX
6    XX    XX
7      XXXX  XX
8
9
A
```

ASCII VALUE = 27H

```
 0 1 2 3 4 5 6 7
0
1          XX
2        XX
3
4
5
6
7
8
9
A
```

ASCII VALUE = 28H

```
 0 1 2 3 4 5 6 7
0
1            XX
2          XX
3        XX
4        XX
5        XX
6          XX
7            XX
8
9
A
```

ASCII VALUE = 29H

```
 0 1 2 3 4 5 6 7
0
1    XX
2      XX
3        XX
4        XX
5        XX
6      XX
7    XX
8
9
A
```

ASCII VALUE = 2AH

```
 0 1 2 3 4 5 6 7
0
1        XX
2  XX    XX    XX
3    XX  XX  XX
4      XXXXXX
5    XX  XX  XX
6  XX    XX    XX
7        XX
8
9
A
```

ASCII VALUE = 2BH

```
 0 1 2 3 4 5 6 7
0
1
2        XX
3        XX
4    XXXXXXXXXX
5        XX
6        XX
7
8
9
A
```

ASCII VALUE = 2CH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6      XXXX
7      XXXX
8        XX
9      XX
A
```

ASCII VALUE = 2DH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4    XXXXXXXXXX
5
6
7
8
9
A
```

ASCII VALUE = 2EH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6      XXXX
7      XXXX
8
9
A
```

ASCII VALUE = 2FH

```
 0 1 2 3 4 5 6 7
0
1              XX
2            XX
3          XX
4        XX
5      XX
6    XX
7  XX
8
9
A
```

ASCII VALUE = 30H

```
 0 1 2 3 4 5 6 7
0
1     XXXXXX
2   XX     XX
3   XX    XXXX
4   XX  XX  XX
5   XXXX   XX
6   XX     XX
7     XXXXXX
8
9
A
```

ASCII VALUE = 31H

```
 0 1 2 3 4 5 6 7
0
1       XX
2     XXXX
3       XX
4       XX
5       XX
6       XX
7     XXXXXX
8
9
A
```

ASCII VALUE = 32H

```
 0 1 2 3 4 5 6 7
0
1     XXXXXX
2   XX     XX
3          XX
4        XX
5     XXXX
6   XX
7   XXXXXXXXXX
8
9
A
```

ASCII VALUE = 33H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXXXXXX
2         XX
3       XX
4     XXXX
5         XX
6   XX    XX
7     XXXXXX
8
9
A
```

ASCII VALUE = 34H

```
 0 1 2 3 4 5 6 7
0
1       XX
2     XX
3   XX
4   XX   XX
5   XXXXXXXXXX
6       XX
7       XX
8
9
A
```

ASCII VALUE = 35H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXXXXXX
2   XX
3   XXXXXXXX
4           XX
5           XX
6   XX      XX
7     XXXXXX
8
9
A
```

ASCII VALUE = 36H

```
 0 1 2 3 4 5 6 7
0
1       XXXX
2     XX
3   XX
4   XXXXXXXX
5   XX      XX
6   XX      XX
7     XXXXXX
8
9
A
```

ASCII VALUE = 37H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXXXXXX
2           XX
3         XX
4       XX
5     XX
6   XX
7   XX
8
9
A
```

ASCII VALUE = 38H

```
 0 1 2 3 4 5 6 7
0
1     XXXXXX
2   XX     XX
3   XX     XX
4     XXXXXX
5   XX     XX
6   XX     XX
7     XXXXXX
8
9
A
```

ASCII VALUE = 39H

```
 0 1 2 3 4 5 6 7
0
1     XXXXXX
2   XX     XX
3   XX     XX
4     XXXXXXX
5           XX
6         XX
7     XXXX
8
9
A
```

ASCII VALUE = 3AH

```
 0 1 2 3 4 5 6 7
0
1
2     XXXX
3     XXXX
4
5
6     XXXX
7     XXXX
8
9
A
```

ASCII VALUE = 3BH

```
 0 1 2 3 4 5 6 7
0
1
2     XXXX
3     XXXX
4
5
6     XXXX
7     XXXX
8       XX
9     XX
A
```

ASCII VALUE = 3CH

```
 0 1 2 3 4 5 6 7
0
1         XX
2       XX
3     XX
4   XX
5     XX
6       XX
7         XX
8
9
A
```

ASCII VALUE = 3DH

```
 0 1 2 3 4 5 6 7
0
1
2
3   XXXXXXXXXX
4
5   XXXXXXXXXX
6
7
8
9
A
```

ASCII VALUE = 3EH

```
 0 1 2 3 4 5 6 7
0
1     XX
2       XX
3         XX
4           XX
5         XX
6       XX
7     XX
8
9
A
```

ASCII VALUE = 3FH

```
 0 1 2 3 4 5 6 7
0
1     XXXXXX
2   XX     XX
3         XX
4       XX
5       XX
6
7       XX
8
9
A
```

ASCII VALUE = 40H

```
 0 1 2 3 4 5 6 7
0
1     XXXXXXXX
2   XX       XX
3 XX   XXXX  XX
4 XX  XX  XX XX
5 XX  XX  XX XX
6 XX  XXXXXXX
7 XX
8   XXXXXXXXX
9
A
```

ASCII VALUE = 41H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXXXX
2 XX       XX
3 XX       XX
4 XX       XX
5 XXXXXXXXXX
6 XX       XX
7 XX       XX
8
9
A
```

ASCII VALUE = 42H

```
 0 1 2 3 4 5 6 7
0
1 XXXXXXXXX
2 XX       XX
3 XX       XX
4 XXXXXXXXX
5 XX       XX
6 XX       XX
7 XXXXXXXXX
8
9
A
```

ASCII VALUE = 43H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXXXX
2 XX       XX
3 XX
4 XX
5 XX
6 XX       XX
7   XXXXXXX
8
9
A
```

ASCII VALUE = 44H

```
 0 1 2 3 4 5 6 7
0
1 XXXXXXXXX
2 XX       XX
3 XX       XX
4 XX       XX
5 XX       XX
6 XX       XX
7 XXXXXXXXX
8
9
A
```

ASCII VALUE = 45H

```
 0 1 2 3 4 5 6 7
0
1 XXXXXXXXXX
2 XX
3 XX
4 XXXXXXXXX
5 XX
6 XX
7 XXXXXXXXXX
8
9
A
```

ASCII VALUE = 46H

```
 0 1 2 3 4 5 6 7
0
1 XXXXXXXXXX
2 XX
3 XX
4 XXXXXXXXX
5 XX
6 XX
7 XX
8
9
A
```

ASCII VALUE = 47H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXXX
2 XX       XX
3 XX
4 XX  XXXXXXX
5 XX       XX
6 XX       XX
7   XXXXXXX
8
9
A
```

ASCII VALUE = 48H

```
 0 1 2 3 4 5 6 7
0
1 XX        XX
2 XX        XX
3 XX        XX
4 XXXXXXXXXXX
5 XX        XX
6 XX        XX
7 XX        XX
8
9
A
```

ASCII VALUE = 49H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXXXXX
2      XX
3      XX
4      XX
5      XX
6      XX
7   XXXXXXXXX
8
9
A
```

ASCII VALUE = 4AH

```
 0 1 2 3 4 5 6 7
0
1   XXXXXXXXX
2         XX
3         XX
4         XX
5         XX
6   XX    XX
7     XXXXX
8
9
A
```

ASCII VALUE = 4BH

```
 0 1 2 3 4 5 6 7
0
1 XX        XX
2 XX      XX
3 XX    XX
4 XXXXX
5 XX    XX
6 XX      XX
7 XX        XX
8
9
A
```

ASCII VALUE = 4CH

```
 0 1 2 3 4 5 6 7
0
1 XX
2 XX
3 XX
4 XX
5 XX
6 XX
7 XXXXXXXXXX
8
9
A
```

ASCII VALUE = 4DH

```
 0 1 2 3 4 5 6 7
0
1 XX          XX
2 XXX       XXX
3 XX XX   XX XX
4 XX    XX    XX
5 XX          XX
6 XX          XX
7 XX          XX
8
9
A
```

ASCII VALUE = 4EH

```
 0 1 2 3 4 5 6 7
0
1 XX        XX
2 XXX       XX
3 XX XX     XX
4 XX   XX   XX
5 XX     XXX
6 XX        XX
7 XX        XX
8
9
A
```

ASCII VALUE = 4FH

```
 0 1 2 3 4 5 6 7
0
1   XXXXXXX
2 XX        XX
3 XX        XX
4 XX        XX
5 XX        XX
6 XX        XX
7   XXXXXXX
8
9
A
```

ASCII VALUE = 50H

```
 0 1 2 3 4 5 6 7
0
1 XXXXXXXXX
2 XX        XX
3 XX        XX
4 XXXXXXXXX
5 XX
6 XX
7 XX
8
9
A
```

ASCII VALUE = 51H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXXX
2 XX        XX
3 XX        XX
4 XX        XX
5 XX    XX  XX
6 XX      XX
7   XXXXX  XX
8
9
A
```

ASCII VALUE = 52H

```
 0 1 2 3 4 5 6 7
0
1 XXXXXXXXX
2 XX        XX
3 XX        XX
4 XXXXXXXXX
5 XX    XX
6 XX      XX
7 XX        XX
8
9
A
```

ASCII VALUE = 53H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXXX
2 XX        XX
3 XX
4   XXXXXXX
5           XX
6 XX        XX
7   XXXXXXX
8
9
A
```

ASCII VALUE = 54H

```
 0 1 2 3 4 5 6 7
0
1 XXXXXXXXXXXXX
2       XX
3       XX
4       XX
5       XX
6       XX
7       XX
8
9
A
```

ASCII VALUE = 55H

```
 0 1 2 3 4 5 6 7
0
1 XX        XX
2 XX        XX
3 XX        XX
4 XX        XX
5 XX        XX
6 XX        XX
7   XXXXXXX
8
9
A
```

ASCII VALUE = 56H

```
 0 1 2 3 4 5 6 7
0
1 XX          XX
2 XX          XX
3 XX          XX
4   XX      XX
5   XX      XX
6     XX  XX
7       XX
8
9
A
```

ASCII VALUE = 57H

```
 0 1 2 3 4 5 6 7
0
1 XX          XX
2 XX          XX
3 XX          XX
4 XX    XX    XX
5 XX  XX  XX  XX
6 XXX       XXX
7 XX          XX
8
9
A
```

ASCII VALUE = 58H

```
 0 1 2 3 4 5 6 7
0
1 XX          XX
2   XX      XX
3     XX  XX
4       XX
5     XX  XX
6   XX      XX
7 XX          XX
8
9
A
```

ASCII VALUE = 59H

```
 0 1 2 3 4 5 6 7
0
1 XX          XX
2 XX          XX
3   XX      XX
4     XX  XX
5       XX
6       XX
7       XX
8
9
A
```

ASCII VALUE = 5AH

```
 0 1 2 3 4 5 6 7
0
1 XXXXXXXXXXX
2           XX
3         XX
4       XX
5     XX
6   XX
7 XXXXXXXXXXX
8
9
A
```

ASCII VALUE = 5BH

```
 0 1 2 3 4 5 6 7
0     XXXXXXX
1     XX
2     XX
3     XX
4     XX
5     XX
6     XX
7     XX
8     XXXXXXX
9
A
```

ASCII VALUE = 5CH

```
 0 1 2 3 4 5 6 7
0
1 XX
2   XX
3     XX
4       XX
5         XX
6           XX
7             XX
8
9
A
```

ASCII VALUE = 5DH

```
 0 1 2 3 4 5 6 7
0 XXXXXXX
1       XX
2       XX
3       XX
4       XX
5       XX
6       XX
7       XX
8 XXXXXXX
9
A
```

ASCII VALUE = 5EH

```
 0 1 2 3 4 5 6 7
0
1       XX
2     XXXXX
3   XXXXXXXXX
4       XX
5       XX
6       XX
7       XX
8       XX
9
A
```

ASCII VALUE = 5FH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6
7
8
9XXXXXXXXXXXXXXX
A
```

ASCII VALUE = 60H

```
 0 1 2 3 4 5 6 7
0   XX
1     XX
2       XX
3
4
5
6
7
8
9
A
```

ASCII VALUE = 61H

```
 0 1 2 3 4 5 6 7
0
1
2
3     XXXXXXXX
4   XX       XX
5   XX       XX
6   XX       XX
7     XXXXXXXXXX
8
9
A
```

ASCII VALUE = 62H

```
 0 1 2 3 4 5 6 7
0
1   XX
2   XX
3   XXXXXXXX
4   XX       XX
5   XX       XX
6   XX       XX
7  XXXXXXXXXX
8
9
A
```

ASCII VALUE = 63H

```
 0 1 2 3 4 5 6 7
0
1
2
3     XXXXXX
4   XX       XX
5   XX
6   XX
7     XXXXXXXX
8
9
A
```

ASCII VALUE = 64H

```
 0 1 2 3 4 5 6 7
0
1           XX
2           XX
3     XXXXXXXX
4   XX       XX
5   XX       XX
6   XX       XX
7     XXXXXXXXXX
8
9
A
```

ASCII VALUE = 65H

```
 0 1 2 3 4 5 6 7
0
1
2
3     XXXXXX
4   XX       XX
5   XXXXXXXX
6   XX
7     XXXXXXXX
8
9
A
```

ASCII VALUE = 66H

```
 0 1 2 3 4 5 6 7
0
1       XXXX
2     XX
3     XX
4   XXXXXX
5     XX
6     XX
7     XX
8
9
A
```

ASCII VALUE = 67H

```
 0 1 2 3 4 5 6 7
0
1
2
3     XXXXXXXX
4   XX       XX
5   XX       XX
6   XX       XX
7     XXXXXXXX
8           XX
9   XXXXXXXX
A
```

ASCII VALUE = 68H

```
 0 1 2 3 4 5 6 7
0
1   XX
2   XX
3   XXXXXXXX
4   XX       XX
5   XX       XX
6   XX       XX
7   XX       XX
8
9
A
```

ASCII VALUE = 69H

```
 0 1 2 3 4 5 6 7
0
1       XX
2
3     XXXX
4       XX
5       XX
6       XX
7     XXXXXX
8
9
A
```

ASCII VALUE = 6AH

```
 0 1 2 3 4 5 6 7
0
1       XX
2
3     XXXX
4       XX
5       XX
6       XX
7       XX
8       XX
9   XXXX
A
```

ASCII VALUE = 6BH

```
 0 1 2 3 4 5 6 7
0
1   XX
2   XX
3   XX    XX
4   XX  XX
5   XXXX
6   XX  XX
7   XX    XX
8
9
A
```

ASCII VALUE = 6CH

```
 0 1 2 3 4 5 6 7
0
1     XXXX
2       XX
3       XX
4       XX
5       XX
6       XX
7     XXXXXX
8
9
A
```

ASCII VALUE = 6DH

```
 0 1 2 3 4 5 6 7
0
1
2
3  XXXXXX  XXXX
4  XX    XX    XX
5  XX    XX    XX
6  XX    XX    XX
7  XX    XX    XX
8
9
A
```

ASCII VALUE = 6EH

```
 0 1 2 3 4 5 6 7
0
1
2
3   XXXXXXXX
4   XX       XX
5   XX       XX
6   XX       XX
7   XX       XX
8
9
A
```

ASCII VALUE = 6FH

```
 0 1 2 3 4 5 6 7
0
1
2
3     XXXXXX
4   XX       XX
5   XX       XX
6   XX       XX
7     XXXXXX
8
9
A
```

ASCII VALUE = 70H

```
 0 1 2 3 4 5 6 7
0
1
2
3   XXXXXXXX
4   XX       XX
5   XX       XX
6   XX       XX
7   XXXXXXXX
8   XX
9   XX
A
```

ASCII VALUE = 71H

```
 0 1 2 3 4 5 6 7
0
1
2
3     XXXXXXXX
4   XX       XX
5   XX       XX
6   XX       XX
7     XXXXXXXX
8           XX
9           XX
A
```

ASCII VALUE = 72H

```
 0 1 2 3 4 5 6 7
0
1
2
3   XX  XXXXXX
4   XXXX
5   XX
6   XX
7   XX
8
9
A
```

ASCII VALUE = 73H

```
 0 1 2 3 4 5 6 7
0
1
2
3     XXXXXXXX
4   XX
5     XXXXXX
6           XX
7   XXXXXXXX
8
9
A
```

ASCII VALUE = 74H

```
 0 1 2 3 4 5 6 7
0
1     XX
2     XX
3   XXXXXXXX
4     XX
5     XX
6     XX
7     XXXXXX
8
9
A
```

ASCII VALUE = 75H

```
 0 1 2 3 4 5 6 7
0
1
2
3   XX      XX
4   XX      XX
5   XX      XX
6   XX      XX
7     XXXXXXXX
8
9
A
```

ASCII VALUE = 76H

```
 0 1 2 3 4 5 6 7
0
1
2
3   XX      XX
4   XX      XX
5   XX      XX
6     XX  XX
7       XX
8
9
A
```

ASCII VALUE = 77H

```
 0 1 2 3 4 5 6 7
0
1
2
3 XX          XX
4 XX          XX
5 XX    XX    XX
6 XX    XX    XX
7   XXXX  XXXX
8
9
A
```

ASCII VALUE = 78H

```
 0 1 2 3 4 5 6 7
0
1
2
3   XX      XX
4     XX  XX
5       XX
6     XX  XX
7   XX      XX
8
9
A
```

ASCII VALUE = 79H

```
 0 1 2 3 4 5 6 7
0
1
2
3   XX      XX
4   XX      XX
5   XX      XX
6     XX  XX
7       XX
8     XX
9   XX
A
```

ASCII VALUE = 7AH

```
 0 1 2 3 4 5 6 7
0
1
2
3   XXXXXXXXXX
4         XX
5       XX
6     XX
7   XXXXXXXXXX
8
9
A
```

ASCII VALUE = 7BH

```
 0 1 2 3 4 5 6 7
0         XXXX
1       XX
2       XX
3     XX
4   XXXX
5     XX
6       XX
7       XX
8         XXXX
9
A
```

ASCII VALUE = 7CH

```
 0 1 2 3 4 5 6 7
0       XX
1       XX
2       XX
3       XX
4       XX
5       XX
6       XX
7       XX
8       XX
9
A
```

ASCII VALUE = 7DH

```
 0 1 2 3 4 5 6 7
0 XXXX
1     XX
2     XX
3       XX
4       XXXX
5       XX
6     XX
7     XX
8 XXXX
9
A
```

ASCII VALUE = 7EH

```
 0 1 2 3 4 5 6 7
0
1   XXXX
2 XX    XX    XX
3         XXXX
4
5
6
7
8
9
A
```

ASCII VALUE = 7FH

```
 0 1 2 3 4 5 6 7
0XX  XX  XX  XX
1 XX  XX  XX  XX
2XX  XX  XX  XX
3 XX  XX  XX  XX
4XX  XX  XX  XX
5 XX  XX  XX  XX
6XX  XX  XX  XX
7 XX  XX  XX  XX
8XX  XX  XX  XX
9 XX  XX  XX  XX
AXX  XX  XX  XX
```

ASCII VALUE = 80H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6
7
8
9
A
```

ASCII VALUE = 81H

```
 0 1 2 3 4 5 6 7
0XXXXXXXXXXXXXXX
1XXXXXXXXXXXXXXX
2XXXXXXXXXXXXXXX
3XXXXXXXXXXXXXXX
4XXXXXXXXXXXXXXX
5XXXXXXXXXXXXXXX
6XXXXXXXXXXXXXXX
7XXXXXXXXXXXXXXX
8XXXXXXXXXXXXXXX
9XXXXXXXXXXXXXXX
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 82H

```
 0 1 2 3 4 5 6 7
0XXXXXXXXXXXXXXX
1XXXXXXXXXXXXXXX
2XXXXXXXXXXXXXXX
3XXXXXXXXXXXXXXX
4XXXXXXXXXXXXXXX
5XXXXXXXXXXXXXXX
6XXXXXXXXXXXXXXX
7XXXXXXXXXXXXXXX
8XXXXXXXXXXXXXXX
9XXXXXXXXXXXXXXX
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 83H

```
 0 1 2 3 4 5 6 7
0XXXXXXXXXXXXXXX
1XXXXXXXXXXXXXXX
2XXXXXXXXXXXXXXX
3XXXXXXXXXXXXXXX
4XXXXXXXXXXXXXXX
5XXXXXXXXXXXXXXX
6XXXXXXXXXXXXXXX
7XXXXXXXXXXXXXXX
8XXXXXXXXXXXXXXX
9XXXXXXXXXXXXXXX
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 84H

```
 0 1 2 3 4 5 6 7
0XXXXXXXXXXXXXXX
1XXXXXXXXXXXXXXX
2XXXXXXXXXXXXXXX
3XXXXXXXXXXXXXXX
4XXXXXXXXXXXXXXX
5XXXXXXXXXXXXXXX
6XXXXXXXXXXXXXXX
7XXXXXXXXXXXXXXX
8XXXXXXXXXXXXXXX
9XXXXXXXXXXXXXXX
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 85H

```
 0 1 2 3 4 5 6 7
0XXXXXXXXXXXXXXX
1XXXXXXXXXXXXXXX
2XXXXXXXXXXXXXXX
3XXXXXXXXXXXXXXX
4XXXXXXXXXXXXXXX
5XXXXXXXXXXXXXXX
6XXXXXXXXXXXXXXX
7XXXXXXXXXXXXXXX
8XXXXXXXXXXXXXXX
9XXXXXXXXXXXXXXX
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 86H

```
 0 1 2 3 4 5 6 7
0 XXXXXXXXXXXXXX
1 XX          XX
2   XX
3     XX
4       XX
5     XX
6   XX
7 XX          XX
8 XXXXXXXXXXXXXX
9
A
```

ASCII VALUE = 87H

```
 0 1 2 3 4 5 6 7
0       XXXXXXXX
1     XX
2     XX
3     XX
4     XX
5     XX
6     XX
7     XX
8     XX
9XXXXXX
A
```

ASCII VALUE = 88H

```
 0 1 2 3 4 5 6 7
0       XXXXXXXX
1     XX
2     XX
3     XX
4     XX
5XXXX XX
6 XX  XX
7   XXXX
8   XXXX
9     XX
A
```

ASCII VALUE = 89H

```
 0 1 2 3 4 5 6 7
0     XXXXXX
1   XX
2     XXXX
3   XX    XX
4     XX    XX
5       XXXX
6           XX
7     XXXXXX
8
9
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 8AH

```
 0 1 2 3 4 5 6 7
0
1
2
3   XX      XX
4   XX      XX
5   XX      XX
6   XX      XX
7   XXXXXXXX  XX
8   XX
9 XX
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 8BH

```
 0 1 2 3 4 5 6 7
0 XX        XX
1     XXXX
2   XX    XX
3 XX        XX
4 XX        XX
5 XXXXXXXXXXX
6 XX        XX
7 XX        XX
8
9
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 8CH

```
 0 1 2 3 4 5 6 7
0 XX        XX
1
2   XXXXXXXX
3 XX        XX
4 XX        XX
5 XX        XX
6 XX        XX
7   XXXXXXXX
8
9
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 8DH

```
 0 1 2 3 4 5 6 7
0 XX        XX
1
2 XX        XX
3 XX        XX
4 XX        XX
5 XX        XX
6 XX        XX
7   XXXXXXXX
8
9
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 8EH

```
 0 1 2 3 4 5 6 7
0       XXXX
1     XX    XX
2     XX    XX
3       XXXX
4
5
6
7
8
9
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 8FH

```
 0 1 2 3 4 5 6 7
0     XXXX
1   XX    XX
2         XX
3     XXXX
4   XX
5   XXXXXXX
6
7
8
9
AXXXXXXXXXXXXXXX
```

CHARACTER GENERATION PROGRAM FOR JK82 VIDEO I VS 1.4 SET: Z96 V.2 25.04.85 16:38:45

ASCII VALUE = 90H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4XXXXXXXXXXXXXXXX
5
6XXXXXXXXXXXXXXXX
7
8
9
A
```

ASCII VALUE = 91H

```
 0 1 2 3 4 5 6 7
0   XX  XX
1   XX  XX
2   XX  XX
3   XX  XX
4   XX  XX
5   XX  XX
6   XX  XX
7   XX  XX
8   XX  XX
9   XX  XX
A   XX  XX
```

ASCII VALUE = 92H

```
 0 1 2 3 4 5 6 7
0   XX  XX
1   XX  XX
2   XX  XX
3   XX  XX
4XXXXXXXXXXXXXXXX
5
6XXXXXXXXXXXXXXXX
7   XX  XX
8   XX  XX
9   XX  XX
A   XX  XX
```

ASCII VALUE = 93H

```
 0 1 2 3 4 5 6 7
0   XX  XX
1   XX  XX
2   XX  XX
3   XX  XX
4   XX  XXXXXX
5   XX
6   XXXXXXXXXXXX
7
8
9
A
```

ASCII VALUE = 94H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4XXXXXXXXXXXX
5           XX
6XXXXXX     XX
7   XX      XX
8   XX      XX
9   XX      XX
A   XX      XX
```

ASCII VALUE = 95H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4   XXXXXXXXXXXX
5   XX
6   XX      XXXXXX
7   XX      XX
8   XX      XX
9   XX      XX
A   XX      XX
```

ASCII VALUE = 96H

```
 0 1 2 3 4 5 6 7
0   XX      XX
1   XX      XX
2   XX      XX
3   XX      XX
4XXXXXX     XX
5           XX
6XXXXXXXXXXXX
7
8
9
A
```

ASCII VALUE = 97H

```
 0 1 2 3 4 5 6 7
0   XX      XX
1   XX      XX
2   XX      XX
3   XX      XX
4   XX      XXXXXX
5   XX
6   XX      XXXXXX
7   XX      XX
8   XX      XX
9   XX      XX
A   XX      XX
```

ASCII VALUE = 98H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4XXXXXXXXXXXXXXXX
5
6XXXXXX     XXXXXX
7   XX      XX
8   XX      XX
9   XX      XX
A   XX      XX
```

ASCII VALUE = 99H

```
 0 1 2 3 4 5 6 7
0   XX      XX
1   XX      XX
2   XX      XX
3   XX      XX
4XXXXXX     XX
5           XX
6XXXXXX     XX
7   XX      XX
8   XX      XX
9   XX      XX
A   XX      XX
```

ASCII VALUE = 9AH

```
 0 1 2 3 4 5 6 7
0   XX      XX
1   XX      XX
2   XX      XX
3   XX      XX
4XXXXXX     XXXXXX
5
6XXXXXXXXXXXXXXXX
7
8
9
A
```

ASCII VALUE = 9BH

```
 0 1 2 3 4 5 6 7
0
1   XX      XX
2
3     XXXXXXXX
4   XX      XX
5   XX      XX
6   XX      XX
7     XXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = 9CH

```
 0 1 2 3 4 5 6 7
0
1   XX      XX
2
3     XXXXXX
4   XX      XX
5   XX      XX
6   XX      XX
7     XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = 9DH

```
 0 1 2 3 4 5 6 7
0
1   XX      XX
2         .
3   XX      XX
4   XX      XX
5   XX      XX
6   XX      XX
7     XXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = 9EH

```
 0 1 2 3 4 5 6 7
0
1     XXXXXX
2   XX      XX
3   XX      XX
4   XX  XXXX
5   XX      XX
6   XX      XX
7   XX  XXXX
8   XX
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = 9FH

```
 0 1 2 3 4 5 6 7
0     XXXX
1   XX    XX
2       XX
3         XX
4   XX    XX
5     XXXX
6
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A0H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A1H

```
 0 1 2 3 4 5 6 7
0
1       XX
2       XX
3       XX
4       XX
5       XX
6
7       XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A2H

```
 0 1 2 3 4 5 6 7
0
1     XX  XX
2     XX  XX
3
4
5
6
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A3H

```
 0 1 2 3 4 5 6 7
0
1     XX  XX
2     XX  XX
3   XXXXXXXXXX
4     XX  XX
5   XXXXXXXXXX
6     XX  XX
7     XX  XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A4H

```
 0 1 2 3 4 5 6 7
0
1       XX
2     XXXXXXXX
3   XX  XX
4     XXXXXX
5       XX  XX
6   XXXXXXXX
7       XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A5H

```
 0 1 2 3 4 5 6 7
0
1   XXXX      XX
2   XXXX    XX
3         XX
4       XX
5     XX
6   XX    XXXX
7 XX      XXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A6H

```
 0 1 2 3 4 5 6 7
0
1     XX
2   XX  XX
3   XX  XX
4     XX
5   XX  XX  XX
6   XX    XX
7     XXXX  XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A7H

```
 0 1 2 3 4 5 6 7
0
1         XX
2       XX
3
4
5
6
7
8
9
AXXXXXXXXXXXXXXXX
```

CHARACTER GENERATION PROGRAM FOR JK82 VIDEO I VS 1.4 SET: Z96 V.2 25.04.85 16:38:52

ASCII VALUE = A8H

```
 0 1 2 3 4 5 6 7
0
1           XX
2         XX
3       XX
4       XX
5       XX
6         XX
7           XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A9H

```
 0 1 2 3 4 5 6 7
0
1   XX
2     XX
3       XX
4       XX
5       XX
6     XX
7   XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = AAH

```
 0 1 2 3 4 5 6 7
0
1       XX
2 XX    XX    XX
3   XX  XX  XX
4     XXXXXX
5   XX  XX  XX
6 XX    XX    XX
7       XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = ABH

```
 0 1 2 3 4 5 6 7
0
1
2       XX
3       XX
4   XXXXXXXXXX
5       XX
6       XX
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = ACH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6     XXXX
7     XXXX
8       XX
9     XX
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = ADH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4   XXXXXXXXXX
5
6
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = AEH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6     XXXX
7     XXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = AFH

```
 0 1 2 3 4 5 6 7
0
1             XX
2           XX
3         XX
4       XX
5     XX
6   XX
7 XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B0H

```
 0 1 2 3 4 5 6 7
0
1     XXXXXX
2   XX      XX
3   XX    XXXX
4   XX  XX  XX
5   XXXX    XX
6   XX      XX
7     XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B1H

```
 0 1 2 3 4 5 6 7
0
1       XX
2     XXXX
3       XX
4       XX
5       XX
6       XX
7     XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B2H

```
 0 1 2 3 4 5 6 7
0
1     XXXXXX
2   XX      XX
3           XX
4         XX
5     XXXX
6   XX
7   XXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B3H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXXXXXX
2           XX
3         XX
4       XXXX
5           XX
6   XX      XX
7     XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B4H

```
 0 1 2 3 4 5 6 7
0
1         XX
2       XX
3     XX
4   XX    XX
5   XXXXXXXXXX
6         XX
7         XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B5H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXXXXXX
2   XX
3   XXXXXXXX
4           XX
5           XX
6   XX      XX
7     XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B6H

```
 0 1 2 3 4 5 6 7
0
1       XXXX
2     XX
3   XX
4   XXXXXXXX
5   XX      XX
6   XX      XX
7     XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B7H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXXXXXX
2           XX
3         XX
4       XX
5     XX
6   XX
7   XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B8H

```
 0 1 2 3 4 5 6 7
0
1     XXXXXX
2   XX      XX
3   XX      XX
4     XXXXXX
5   XX      XX
6   XX      XX
7     XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B9H

```
 0 1 2 3 4 5 6 7
0
1     XXXXXX
2   XX      XX
3   XX      XX
4     XXXXXXX
5           XX
6         XX
7     XXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = BAH

```
 0 1 2 3 4 5 6 7
0
1
2     XXXX
3     XXXX
4
5
6     XXXX
7     XXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = BBH

```
 0 1 2 3 4 5 6 7
0
1
2     XXXX
3     XXXX
4
5
6     XXXX
7     XXXX
8       XX
9     XX
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = BCH

```
 0 1 2 3 4 5 6 7
0
1         XX
2       XX
3     XX
4   XX
5     XX
6       XX
7         XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = BDH

```
 0 1 2 3 4 5 6 7
0
1
2
3   XXXXXXXXXX
4
5   XXXXXXXXXX
6
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = BEH

```
 0 1 2 3 4 5 6 7
0
1     XX
2       XX
3         XX
4           XX
5         XX
6       XX
7     XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = BFH

```
 0 1 2 3 4 5 6 7
0
1     XXXXXX
2   XX      XX
3         XX
4       XX
5       XX
6
7       XX
8
9
AXXXXXXXXXXXXXXXX
```

CHARACTER GENERATION PROGRAM FOR JK82 VIDEO I VS 1.4 SET: Z96 V.2 25.04.85 16:39:00

ASCII VALUE = C0H

```
 0 1 2 3 4 5 6 7
0
1      XXXXXXXX
2    XX        XX
3  XX    XXXX  XX
4  XX  XX  XX  XX
5  XX  XX  XX  XX
6  XX  XXXXXXXX
7  XX
8    XXXXXXXXXX
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C1H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXXXX
2  XX        XX
3  XX        XX
4  XX        XX
5  XXXXXXXXXXXX
6  XX        XX
7  XX        XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C2H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2  XX        XX
3  XX        XX
4  XXXXXXXXXX
5  XX        XX
6  XX        XX
7  XXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C3H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXXXX
2  XX        XX
3  XX
4  XX
5  XX
6  XX        XX
7    XXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C4H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2  XX        XX
3  XX        XX
4  XX        XX
5  XX        XX
6  XX        XX
7  XXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C5H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXXXX
2  XX
3  XX
4  XXXXXXXXXX
5  XX
6  XX
7  XXXXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C6H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXXXX
2  XX
3  XX
4  XXXXXXXXXX
5  XX
6  XX
7  XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C7H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXXXX
2  XX        XX
3  XX
4  XX  XXXXXXX
5  XX        XX
6  XX        XX
7    XXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C8H

```
 0 1 2 3 4 5 6 7
0
1  XX        XX
2  XX        XX
3  XX        XX
4  XXXXXXXXXXXX
5  XX        XX
6  XX        XX
7  XX        XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C9H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXXXXXX
2        XX
3        XX
4        XX
5        XX
6        XX
7    XXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = CAH

```
 0 1 2 3 4 5 6 7
0
1    XXXXXXXXXX
2            XX
3            XX
4            XX
5            XX
6    XX      XX
7      XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = CBH

```
 0 1 2 3 4 5 6 7
0
1  XX        XX
2  XX      XX
3  XX    XX
4  XXXXXX
5  XX    XX
6  XX      XX
7  XX        XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = CCH

```
 0 1 2 3 4 5 6 7
0
1  XX
2  XX
3  XX
4  XX
5  XX
6  XX
7  XXXXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = CDH

```
 0 1 2 3 4 5 6 7
0
1  XX          XX
2  XXXX      XXXX
3  XX  XX  XX  XX
4  XX    XX    XX
5  XX          XX
6  XX          XX
7  XX          XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = CEH

```
 0 1 2 3 4 5 6 7
0
1  XX        XX
2  XXXX      XX
3  XX  XX    XX
4  XX    XX  XX
5  XX      XXXX
6  XX        XX
7  XX        XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = CFH

```
 0 1 2 3 4 5 6 7
0
1    XXXXXXXX
2  XX        XX
3  XX        XX
4  XX        XX
5  XX        XX
6  XX        XX
7    XXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D0H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2  XX        XX
3  XX        XX
4  XXXXXXXXXX
5  XX
6  XX
7  XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D1H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXXXX
2  XX        XX
3  XX        XX
4  XX        XX
5  XX    XX  XX
6  XX      XX
7    XXXXXX  XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D2H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2  XX        XX
3  XX        XX
4  XXXXXXXXXX
5  XX    XX
6  XX      XX
7  XX        XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D3H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXXXX
2  XX        XX
3  XX
4    XXXXXXXX
5            XX
6  XX        XX
7    XXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D4H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXXXXXX
2        XX
3        XX
4        XX
5        XX
6        XX
7        XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D5H

```
 0 1 2 3 4 5 6 7
0
1  XX        XX
2  XX        XX
3  XX        XX
4  XX        XX
5  XX        XX
6  XX        XX
7    XXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D6H

```
 0 1 2 3 4 5 6 7
0
1  XX          XX
2  XX          XX
3  XX          XX
4    XX      XX
5    XX      XX
6      XX  XX
7        XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D7H

```
 0 1 2 3 4 5 6 7
0
1  XX          XX
2  XX          XX
3  XX          XX
4  XX    XX    XX
5  XX  XX  XX  XX
6  XXXX      XXXX
7  XX          XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D8H

```
 0 1 2 3 4 5 6 7
0
1  XX          XX
2    XX      XX
3      XX  XX
4        XX
5      XX  XX
6    XX      XX
7  XX          XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D9H

```
 0 1 2 3 4 5 6 7
0
1  XX          XX
2  XX          XX
3    XX      XX
4      XX  XX
5        XX
6        XX
7        XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = DAH

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXXXX
2            XX
3          XX
4        XX
5      XX
6    XX
7  XXXXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = DBH

```
 0 1 2 3 4 5 6 7
0      XXXXXXXX
1      XX
2      XX
3      XX
4      XX
5      XX
6      XX
7      XX
8      XXXXXXXX
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = DCH

```
 0 1 2 3 4 5 6 7
0
1  XX
2    XX
3      XX
4        XX
5          XX
6            XX
7              XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = DDH

```
 0 1 2 3 4 5 6 7
0  XXXXXXXX
1        XX
2        XX
3        XX
4        XX
5        XX
6        XX
7        XX
8  XXXXXXXX
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = DEH

```
 0 1 2 3 4 5 6 7
0
1        XX
2      XXXXXX
3    XXXXXXXXXX
4        XX
5        XX
6        XX
7        XX
8        XX
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = DFH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6
7
8
9XXXXXXXXXXXXXXXX
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E0H

```
 0 1 2 3 4 5 6 7
0      XX
1        XX
2          XX
3
4
5
6
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E1H

```
 0 1 2 3 4 5 6 7
0
1
2
3      XXXXXXXX
4    XX      XX
5    XX      XX
6    XX      XX
7      XXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E2H

```
 0 1 2 3 4 5 6 7
0
1    XX
2    XX
3    XXXXXXXX
4    XX      XX
5    XX      XX
6    XX      XX
7  XXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E3H

```
 0 1 2 3 4 5 6 7
0
1
2
3      XXXXXX
4    XX      XX
5    XX
6    XX
7      XXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E4H

```
 0 1 2 3 4 5 6 7
0
1            XX
2            XX
3      XXXXXXXX
4    XX      XX
5    XX      XX
6    XX      XX
7      XXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E5H

```
 0 1 2 3 4 5 6 7
0
1
2
3      XXXXXX
4    XX      XX
5    XXXXXXXX
6    XX
7      XXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E6H

```
 0 1 2 3 4 5 6 7
0
1        XXXX
2      XX
3      XX
4    XXXXXX
5      XX
6      XX
7      XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E7H

```
 0 1 2 3 4 5 6 7
0
1
2
3      XXXXXXXX
4    XX      XX
5    XX      XX
6    XX      XX
7      XXXXXXXX
8            XX
9    XXXXXXXX
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E8H

```
 0 1 2 3 4 5 6 7
0
1    XX
2    XX
3    XXXXXXXX
4    XX      XX
5    XX      XX
6    XX      XX
7    XX      XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E9H

```
 0 1 2 3 4 5 6 7
0
1        XX
2
3      XXXX
4        XX
5        XX
6        XX
7      XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = EAH

```
 0 1 2 3 4 5 6 7
0
1        XX
2
3      XXXX
4        XX
5        XX
6        XX
7        XX
8        XX
9    XXXX
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = EBH

```
 0 1 2 3 4 5 6 7
0
1    XX
2    XX
3    XX    XX
4    XX  XX
5    XXXX
6    XX  XX
7    XX    XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = ECH

```
 0 1 2 3 4 5 6 7
0
1      XXXX
2        XX
3        XX
4        XX
5        XX
6        XX
7      XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = EDH

```
 0 1 2 3 4 5 6 7
0
1
2
3  XXXXXX  XXXX
4  XX    XX    XX
5  XX    XX    XX
6  XX    XX    XX
7  XX    XX    XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = EEH

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXXXXXX
4    XX      XX
5    XX      XX
6    XX      XX
7    XX      XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = EFH

```
 0 1 2 3 4 5 6 7
0
1
2
3      XXXXXX
4    XX      XX
5    XX      XX
6    XX      XX
7      XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = F0H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXXXXXX
4    XX      XX
5    XX      XX
6    XX      XX
7    XXXXXXXX
8    XX
9    XX
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = F1H

```
 0 1 2 3 4 5 6 7
0
1
2
3      XXXXXXXX
4    XX      XX
5    XX      XX
6    XX      XX
7      XXXXXXXX
8           XX
9           XX
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = F2H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XX  XXXXXX
4    XXXX
5    XX
6    XX
7    XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = F3H

```
 0 1 2 3 4 5 6 7
0
1
2
3      XXXXXXXX
4    XX
5      XXXXXX
6           XX
7    XXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = F4H

```
 0 1 2 3 4 5 6 7
0
1      XX
2      XX
3    XXXXXXXX
4      XX
5      XX
6      XX
7      XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = F5H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XX      XX
4    XX      XX
5    XX      XX
6    XX      XX
7      XXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = F6H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XX      XX
4    XX      XX
5    XX      XX
6      XX  XX
7        XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = F7H

```
 0 1 2 3 4 5 6 7
0
1
2
3  XX          XX
4  XX          XX
5  XX    XX    XX
6  XX    XX    XX
7    XXXX  XXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = F8H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XX      XX
4      XX  XX
5        XX
6      XX  XX
7    XX      XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = F9H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XX      XX
4    XX      XX
5    XX      XX
6      XX  XX
7        XX
8      XX
9    XX
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = FAH

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXXXXXXXX
4          XX
5        XX
6      XX
7    XXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = FBH

```
 0 1 2 3 4 5 6 7
0          XXXX
1        XX
2        XX
3      XX
4    XXXX
5      XX
6        XX
7        XX
8          XXXX
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = FCH

```
 0 1 2 3 4 5 6 7
0        XX
1        XX
2        XX
3        XX
4        XX
5        XX
6        XX
7        XX
8        XX
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = FDH

```
 0 1 2 3 4 5 6 7
0  XXXX
1      XX
2      XX
3        XX
4        XXXX
5        XX
6      XX
7      XX
8  XXXX
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = FEH

```
 0 1 2 3 4 5 6 7
0
1    XXXX
2  XX    XX    XX
3           XXXX
4
5
6
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = FFH

```
 0 1 2 3 4 5 6 7
0XX  XX  XX  XX
1  XX  XX  XX  XX
2XX  XX  XX  XX
3  XX  XX  XX  XX
4XX  XX  XX  XX
5  XX  XX  XX  XX
6XX  XX  XX  XX
7  XX  XX  XX  XX
8XX  XX  XX  XX
9  XX  XX  XX  XX
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = 00H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6
7
8
9
A
```

ASCII VALUE = 01H

```
 0 1 2 3 4 5 6 7
0
1    XX
2  XXXXXX
3XX  XX  XX
4    XX
5    XX
6    XX
7    XX
8    XX
9    XX
A    XX
```

ASCII VALUE = 02H

```
 0 1 2 3 4 5 6 7
0    XX
1    XX
2    XX
3    XX
4    XX
5    XX
6    XX
7XX  XX  XX
8  XXXXXX
9    XX
A
```

ASCII VALUE = 03H

```
 0 1 2 3 4 5 6 7
0
1
2
3      XX
4    XX
5  XXXXXXXXXX
6    XX
7      XX
8
9
A
```

ASCII VALUE = 04H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XX
4      XX
5XXXXXXXXXX
6      XX
7    XX
8
9
A
```

ASCII VALUE = 05H

```
 0 1 2 3 4 5 6 7
0
1XXXXXX
2XXXX
3XX  XX
4      XX
5        XX
6          XX
7
8
9
A
```

ASCII VALUE = 06H

```
 0 1 2 3 4 5 6 7
0
1      XXXXXX
2        XXXX
3      XX  XX
4    XX
5  XX
6XX
7
8
9
A
```

ASCII VALUE = 07H

```
 0 1 2 3 4 5 6 7
0
1
2XX
3  XX
4    XX
5      XX  XX
6        XXXX
7      XXXXXX
8
9
A
```

ASCII VALUE = 08H

```
 0 1 2 3 4 5 6 7
0
1
2           XX
3         XX
4       XX
5XX   XX
6XXXX
7XXXXXX
8
9
A
```

ASCII VALUE = 09H

```
 0 1 2 3 4 5 6 7
0    XXXXXX
1  XX
2    XXXX
3  XX    XX
4    XX    XX
5      XXXX
6          XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 0AH

```
 0 1 2 3 4 5 6 7
0
1
2
3  XX    XX
4  XX    XX
5  XX    XX
6  XX    XX
7  XXXXXX  XX
8  XX
9XX
A
```

ASCII VALUE = 0BH

```
 0 1 2 3 4 5 6 7
0  XX      XX
1      XX
2    XX  XX
3  XX      XX
4  XX      XX
5  XXXXXXXXXX
6  XX      XX
7  XX      XX
8
9
A
```

ASCII VALUE = 0CH

```
 0 1 2 3 4 5 6 7
0  XX      XX
1
2    XXXXXX
3  XX      XX
4  XX      XX
5  XX      XX
6  XX      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 0DH

```
 0 1 2 3 4 5 6 7
0  XX      XX
1
2  XX      XX
3  XX      XX
4  XX      XX
5  XX      XX
6  XX      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 0EH

```
 0 1 2 3 4 5 6 7
0    XXXX
1  XX    XX
2  XX    XX
3    XXXX
4
5
6
7
8
9
A
```

ASCII VALUE = 0FH

```
 0 1 2 3 4 5 6 7
0    XXXX
1  XX    XX
2        XX
3    XXXX
4  XX
5  XXXXXXXX
6
7
8
9
A
```

ASCII VALUE = 10H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5XXXXXXXXXXXXXXXX
6
7
8
9
A
```

ASCII VALUE = 11H

```
 0 1 2 3 4 5 6 7
0    XX
1    XX
2    XX
3    XX
4    XX
5    XX
6    XX
7    XX
8    XX
9    XX
A    XX
```

ASCII VALUE = 12H

```
 0 1 2 3 4 5 6 7
0    XX
1    XX
2    XX
3    XX
4    XX
5XXXXXXXXXXXXXXXX
6    XX
7    XX
8    XX
9    XX
A    XX
```

ASCII VALUE = 13H

```
 0 1 2 3 4 5 6 7
0    XX
1    XX
2    XX
3    XX
4    XX
5    XXXXXXXX
6
7
8
9
A
```

ASCII VALUE = 14H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5XXXXXX
6    XX
7    XX
8    XX
9    XX
A    XX
```

ASCII VALUE = 15H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5    XXXXXXXX
6    XX
7    XX
8    XX
9    XX
A    XX
```

ASCII VALUE = 16H

```
 0 1 2 3 4 5 6 7
0    XX
1    XX
2    XX
3    XX
4    XX
5XXXXXX
6
7
8
9
A
```

ASCII VALUE = 17H

```
 0 1 2 3 4 5 6 7
0    XX
1    XX
2    XX
3    XX
4    XX
5    XXXXXXXX
6    XX
7    XX
8    XX
9    XX
A    XX
```

ASCII VALUE = 18H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5XXXXXXXXXXXXXXXX
6   XX
7   XX
8   XX
9   XX
A   XX
```

ASCII VALUE = 19H

```
 0 1 2 3 4 5 6 7
0   XX
1   XX
2   XX
3   XX
4   XX
5XXXXXX
6   XX
7   XX
8   XX
9   XX
A   XX
```

ASCII VALUE = 1AH

```
 0 1 2 3 4 5 6 7
0   XX
1   XX
2   XX
3   XX
4   XX
5XXXXXXXXXXXXXXXX
6
7
8
9
A
```

ASCII VALUE = 1BH

```
 0 1 2 3 4 5 6 7
0
1 XX    XX
2
3   XXXXXX
4 XX    XX
5 XX    XX
6 XX    XX
7   XXXXXXX
8
9
A
```

ASCII VALUE = 1CH

```
 0 1 2 3 4 5 6 7
0
1 XX    XX
2
3   XXXX
4 XX    XX
5 XX    XX
6 XX    XX
7   XXXX
8
9
A
```

ASCII VALUE = 1DH

```
 0 1 2 3 4 5 6 7
0
1 XX    XX
2
3 XX    XX
4 XX    XX
5 XX    XX
6 XX    XX
7   XXXXXX
8
9
A
```

ASCII VALUE = 1EH

```
 0 1 2 3 4 5 6 7
0
1   XXXX
2 XX    XX
3 XX    XX
4 XX  XX
5 XX    XX
6 XX    XX
7 XX  XX
8 XX
9
A
```

ASCII VALUE = 1FH

```
 0 1 2 3 4 5 6 7
0   XXXX
1 XX    XX
2     XX
3       XX
4 XX    XX
5   XXXX
6
7
8
9
A
```

ASCII VALUE = 20H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6
7
8
9
A
```

ASCII VALUE = 21H

```
 0 1 2 3 4 5 6 7
0
1     XX
2     XX
3     XX
4     XX
5     XX
6
7     XX
8
9
A
```

ASCII VALUE = 22H

```
 0 1 2 3 4 5 6 7
0
1   XX  XX
2   XX  XX
3
4
5
6
7
8
9
A
```

ASCII VALUE = 23H

```
 0 1 2 3 4 5 6 7
0
1   XX  XX
2   XX  XX
3 XXXXXXXXXX
4   XX  XX
5 XXXXXXXXXX
6   XX  XX
7   XX  XX
8
9
A
```

ASCII VALUE = 24H

```
 0 1 2 3 4 5 6 7
0
1     XX
2   XXXXXXXX
3 XX  XX
4   XXXXXX
5     XX  XX
6 XXXXXXXX
7     XX
8
9
A
```

ASCII VALUE = 25H

```
 0 1 2 3 4 5 6 7
0
1 XXXX
2 XXXX    XX
3       XX
4     XX
5   XX
6 XX    XXXX
7       XXXX
8
9
A
```

ASCII VALUE = 26H

```
 0 1 2 3 4 5 6 7
0
1   XX
2 XX  XX
3 XX  XX
4   XX
5 XX  XX  XX
6 XX    XX
7   XXXX  XX
8
9
A
```

ASCII VALUE = 27H

```
 0 1 2 3 4 5 6 7
0
1       XX
2     XX
3
4
5
6
7
8
9
A
```

ASCII VALUE = 28H

```
 0 1 2 3 4 5 6 7
0
1         XX
2       XX
3     XX
4     XX
5     XX
6       XX
7         XX
8
9
A
```

ASCII VALUE = 29H

```
 0 1 2 3 4 5 6 7
0
1 XX
2   XX
3     XX
4     XX
5     XX
6   XX
7 XX
8
9
A
```

ASCII VALUE = 2AH

```
 0 1 2 3 4 5 6 7
0
1
2     XX
3 XX  XX  XX
4   XXXXXX
5 XX  XX  XX
6     XX
7
8
9
A
```

ASCII VALUE = 2BH

```
 0 1 2 3 4 5 6 7
0
1
2     XX
3     XX
4 XXXXXXXXXX
5     XX
6     XX
7
8
9
A
```

ASCII VALUE = 2CH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6   XXXX
7   XXXX
8     XX
9   XX
A
```

ASCII VALUE = 2DH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4 XXXXXXXXXX
5
6
7
8
9
A
```

ASCII VALUE = 2EH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6   XXXX
7   XXXX
8
9
A
```

ASCII VALUE = 2FH

```
 0 1 2 3 4 5 6 7
0
1
2         XX
3       XX
4     XX
5   XX
6 XX
7
8
9
A
```

CHARACTER GENERATION PROGRAM FOR JK82 VIDEO I VS 1.4 SET: Z132 V.2 25.04.85 16:43:57

ASCII VALUE = 30H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX    XXXX
4  XX  XX  XX
5  XXXX    XX
6  XX      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 31H

```
 0 1 2 3 4 5 6 7
0
1      XX
2    XXXX
3      XX
4      XX
5      XX
6      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 32H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3          XX
4        XX
5    XXXX
6  XX
7  XXXXXXXXXX
8
9
A
```

ASCII VALUE = 33H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2          XX
3        XX
4      XXXX
5          XX
6  XX      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 34H

```
 0 1 2 3 4 5 6 7
0
1        XX
2      XX
3    XX
4  XX    XX
5  XXXXXXXXXX
6        XX
7        XX
8
9
A
```

ASCII VALUE = 35H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2  XX
3  XXXXXXXX
4           XX
5           XX
6  XX      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 36H

```
 0 1 2 3 4 5 6 7
0
1      XXXX
2    XX
3  XX
4  XXXXXXXX
5  XX      XX
6  XX      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 37H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2           XX
3         XX
4       XX
5     XX
6     XX
7     XX
8
9
A
```

ASCII VALUE = 38H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX      XX
4    XXXXXX
5  XX      XX
6  XX      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 39H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX      XX
4    XXXXXXX
5           XX
6         XX
7    XXXX
8
9
A
```

ASCII VALUE = 3AH

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXX
4    XXXX
5
6    XXXX
7    XXXX
8
9
A
```

ASCII VALUE = 3BH

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXX
4    XXXX
5
6    XXXX
7    XXXX
8      XX
9    XX
A
```

ASCII VALUE = 3CH

```
 0 1 2 3 4 5 6 7
0
1            XX
2          XX
3        XX
4    XX
5      XX
6        XX
7          XX
8
9
A
```

ASCII VALUE = 3DH

```
 0 1 2 3 4 5 6 7
0
1
2
3  XXXXXXXXXX
4
5  XXXXXXXXXX
6
7
8
9
A
```

ASCII VALUE = 3EH

```
 0 1 2 3 4 5 6 7
0
1  XX
2    XX
3      XX
4        XX
5      XX
6    XX
7  XX
8
9
A
```

ASCII VALUE = 3FH

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3          XX
4      XX
5      XX
6
7      XX
8
9
A
```

ASCII VALUE = 40H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX  XXXXXX
4  XX  XX  XX
5  XX  XXXXXX
6  XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 41H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX      XX
4  XX      XX
5  XXXXXXXXXX
6  XX      XX
7  XX      XX
8
9
A
```

ASCII VALUE = 42H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXX
2  XX      XX
3  XX      XX
4  XXXXXXXX
5  XX      XX
6  XX      XX
7  XXXXXXXX
8
9
A
```

ASCII VALUE = 43H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX
4  XX
5  XX
6  XX      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 44H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXX
2  XX      XX
3  XX      XX
4  XX      XX
5  XX      XX
6  XX      XX
7  XXXXXXXX
8
9
A
```

ASCII VALUE = 45H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2  XX
3  XX
4  XXXXXXXX
5  XX
6  XX
7  XXXXXXXXXX
8
9
A
```

ASCII VALUE = 46H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2  XX
3  XX
4  XXXXXXXX
5  XX
6  XX
7  XX
8
9
A
```

ASCII VALUE = 47H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX
4  XX  XXXXXX
5  XX      XX
6  XX      XX
7    XXXXXX
8
9
A
```

CHARACTER GENERATION PROGRAM FOR JK82 VIDEO I VS 1.4 SET: Z132 V.2 25.04.85 16:45:24

ASCII VALUE = 48H

```
 0 1 2 3 4 5 6 7
0
1  XX       XX
2  XX       XX
3  XX       XX
4  XXXXXXXXXX
5  XX       XX
6  XX       XX
7  XX       XX
8
9
A
```

ASCII VALUE = 49H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2      XX
3      XX
4      XX
5      XX
6      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 4AH

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXX
2         XX
3         XX
4         XX
5         XX
6  XX     XX
7    XXXX
8
9
A
```

ASCII VALUE = 4BH

```
 0 1 2 3 4 5 6 7
0
1  XX       XX
2  XX     XX
3  XX   XX
4  XXXX
5  XX   XX
6  XX     XX
7  XX       XX
8
9
A
```

ASCII VALUE = 4CH

```
 0 1 2 3 4 5 6 7
0
1  XX
2  XX
3  XX
4  XX
5  XX
6  XX
7  XXXXXXXXXX
8
9
A
```

ASCII VALUE = 4DH

```
 0 1 2 3 4 5 6 7
0
1  XX       XX
2  XXXX   XXXX
3  XX  XX   XX
4  XX  XX   XX
5  XX       XX
6  XX       XX
7  XX       XX
8
9
A
```

ASCII VALUE = 4EH

```
 0 1 2 3 4 5 6 7
0
1  XX       XX
2  XXXX     XX
3  XX  XX   XX
4  XX     XXXX
5  XX       XX
6  XX       XX
7  XX       XX
8
9
A
```

ASCII VALUE = 4FH

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX      XX
4  XX      XX
5  XX      XX
6  XX      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 50H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXX
2  XX      XX
3  XX      XX
4  XXXXXXXX
5  XX
6  XX
7  XX
8
9
A
```

ASCII VALUE = 51H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX      XX
4  XX      XX
5  XX  XX  XX
6  XX    XX
7    XXXX  XX
8
9
A
```

ASCII VALUE = 52H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXX
2  XX      XX
3  XX      XX
4  XXXXXXXX
5  XX  XX
6  XX    XX
7  XX      XX
8
9
A
```

ASCII VALUE = 53H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX
4    XXXXXX
5          XX
6  XX      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 54H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2      XX
3      XX
4      XX
5      XX
6      XX
7      XX
8
9
A
```

ASCII VALUE = 55H

```
 0 1 2 3 4 5 6 7
0
1  XX      XX
2  XX      XX
3  XX      XX
4  XX      XX
5  XX      XX
6  XX      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 56H

```
 0 1 2 3 4 5 6 7
0
1  XX      XX
2  XX      XX
3  XX      XX
4    XX  XX
5    XX  XX
6      XX
7      XX
8
9
A
```

ASCII VALUE = 57H

```
 0 1 2 3 4 5 6 7
0
1  XX      XX
2  XX      XX
3  XX      XX
4  XX  XX  XX
5  XX  XX  XX
6  XX  XX  XX
7    XX  XX
8
9
A
```

ASCII VALUE = 58H

```
 0 1 2 3 4 5 6 7
0
1  XX      XX
2  XX      XX
3    XX  XX
4      XX
5    XX  XX
6  XX      XX
7  XX      XX
8
9
A
```

ASCII VALUE = 59H

```
 0 1 2 3 4 5 6 7
0
1  XX      XX
2  XX      XX
3    XX  XX
4      XX
5      XX
6      XX
7      XX
8
9
A
```

ASCII VALUE = 5AH

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2          XX
3        XX
4      XX
5    XX
6  XX
7  XXXXXXXXXX
8
9
A
```

ASCII VALUE = 5BH

```
 0 1 2 3 4 5 6 7
0      XXXXXX
1      XX
2      XX
3      XX
4      XX
5      XX
6      XX
7      XX
8      XXXXXX
9
A
```

ASCII VALUE = 5CH

```
 0 1 2 3 4 5 6 7
0
1
2  XX
3    XX
4      XX
5        XX
6          XX
7
8
9
A
```

ASCII VALUE = 5DH

```
 0 1 2 3 4 5 6 7
0  XXXXXX
1      XX
2      XX
3      XX
4      XX
5      XX
6      XX
7      XX
8  XXXXXX
9
A
```

ASCII VALUE = 5EH

```
 0 1 2 3 4 5 6 7
0
1      XX
2    XXXXXX
3  XXXXXXXXXX
4      XX
5      XX
6      XX
7      XX
8
9
A
```

ASCII VALUE = 5FH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6
7
8
9XXXXXXXXXXXXXXXX
A
```

CHARACTER GENERATION PROGRAM FOR JK82 VIDEO I VS 1.4 SET: Z132 V.2 25.04.85 16:46:52

ASCII VALUE = 60H

```
 0 1 2 3 4 5 6 7
0
1    XX
2      XX
3        XX
4
5
6
7
8
9
A
```

ASCII VALUE = 61H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXXXX
4  XX    XX
5  XX    XX
6  XX    XX
7    XXXXXXXX
8
9
A
```

ASCII VALUE = 62H

```
 0 1 2 3 4 5 6 7
0
1  XX
2  XX
3  XXXXXX
4  XX    XX
5  XX    XX
6  XX    XX
7  XXXXXX
8
9
A
```

ASCII VALUE = 63H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXX
4  XX    XX
5  XX
6  XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 64H

```
 0 1 2 3 4 5 6 7
0
1        XX
2        XX
3    XXXXXX
4  XX    XX
5  XX    XX
6  XX    XX
7    XXXXXXXX
8
9
A
```

ASCII VALUE = 65H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXX
4  XX    XX
5  XXXXXX
6  XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 66H

```
 0 1 2 3 4 5 6 7
0
1      XXXX
2    XX
3    XX
4  XXXXXX
5    XX
6    XX
7    XX
8
9
A
```

ASCII VALUE = 67H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXXXX
4  XX    XX
5  XX    XX
6  XX    XX
7    XXXXXX
8        XX
9  XXXXXX
A
```

ASCII VALUE = 68H

```
 0 1 2 3 4 5 6 7
0
1  XX
2  XX
3  XXXXXX
4  XX    XX
5  XX    XX
6  XX    XX
7  XX    XX
8
9
A
```

ASCII VALUE = 69H

```
 0 1 2 3 4 5 6 7
0
1      XX
2
3    XXXX
4      XX
5      XX
6      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 6AH

```
 0 1 2 3 4 5 6 7
0
1        XX
2
3      XXXX
4        XX
5        XX
6        XX
7        XX
8        XX
9    XXXX
A
```

ASCII VALUE = 6BH

```
 0 1 2 3 4 5 6 7
0
1  XX
2  XX
3  XX    XX
4  XX  XX
5  XXXX
6  XX  XX
7  XX    XX
8
9
A
```

ASCII VALUE = 6CH

```
 0 1 2 3 4 5 6 7
0
1    XXXX
2      XX
3      XX
4      XX
5      XX
6      XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 6DH

```
 0 1 2 3 4 5 6 7
0
1
2
3  XXXX  XX
4  XX  XX  XX
5  XX  XX  XX
6  XX  XX  XX
7  XX  XX  XX
8
9
A
```

ASCII VALUE = 6EH

```
 0 1 2 3 4 5 6 7
0
1
2
3  XXXXXX
4  XX    XX
5  XX    XX
6  XX    XX
7  XX    XX
8
9
A
```

ASCII VALUE = 6FH

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXX
4  XX    XX
5  XX    XX
6  XX    XX
7    XXXX
8
9
A
```

ASCII VALUE = 70H

```
 0 1 2 3 4 5 6 7
0
1
2
3  XXXXXX
4  XX    XX
5  XX    XX
6  XX    XX
7  XXXXXX
8  XX
9  XX
A
```

ASCII VALUE = 71H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXXXX
4  XX    XX
5  XX    XX
6  XX    XX
7    XXXXXX
8        XX
9        XX
A
```

ASCII VALUE = 72H

```
 0 1 2 3 4 5 6 7
0
1
2
3  XX  XXXX
4  XXXX
5  XX
6  XX
7  XX
8
9
A
```

ASCII VALUE = 73H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXXXX
4  XX
5    XXXX
6        XX
7  XXXXXX
8
9
A
```

ASCII VALUE = 74H

```
 0 1 2 3 4 5 6 7
0
1      XX
2      XX
3    XXXXXX
4      XX
5      XX
6      XX
7      XXXX
8
9
A
```

ASCII VALUE = 75H

```
 0 1 2 3 4 5 6 7
0
1
2
3  XX    XX
4  XX    XX
5  XX    XX
6  XX    XX
7    XXXXXX
8
9
A
```

ASCII VALUE = 76H

```
 0 1 2 3 4 5 6 7
0
1
2
3  XX      XX
4  XX      XX
5    XX  XX
6    XX  XX
7      XX
8
9
A
```

ASCII VALUE = 77H

```
 0 1 2 3 4 5 6 7
0
1
2
3  XX      XX
4  XX      XX
5  XX  XX  XX
6  XX  XX  XX
7    XX  XX
8
9
A
```

ASCII VALUE = 78H

```
 0 1 2 3 4 5 6 7
0
1
2
3  XX      XX
4    XX  XX
5      XX
6    XX  XX
7  XX      XX
8
9
A
```

ASCII VALUE = 79H

```
 0 1 2 3 4 5 6 7
0
1
2
3  XX      XX
4  XX      XX
5  XX      XX
6    XX  XX
7      XX
8    XX
9  XX
A
```

ASCII VALUE = 7AH

```
 0 1 2 3 4 5 6 7
0
1
2
3  XXXXXXXXXX
4        XX
5      XX
6    XX
7  XXXXXXXXXX
8
9
A
```

ASCII VALUE = 7BH

```
 0 1 2 3 4 5 6 7
0        XXXX
1        XX
2      XX
3      XX
4    XX
5      XX
6      XX
7        XX
8        XXXX
9
A
```

ASCII VALUE = 7CH

```
 0 1 2 3 4 5 6 7
0      XX
1      XX
2      XX
3      XX
4      XX
5      XX
6      XX
7      XX
8      XX
9
A
```

ASCII VALUE = 7DH

```
 0 1 2 3 4 5 6 7
0  XXXX
1    XX
2      XX
3      XX
4        XX
5      XX
6      XX
7    XX
8  XXXX
9
A
```

ASCII VALUE = 7EH

```
 0 1 2 3 4 5 6 7
0
1    XX
2  XX  XX  XX
3        XX
4
5
6
7
8
9
A
```

ASCII VALUE = 7FH

```
 0 1 2 3 4 5 6 7
0  XX  XX  XX
1XX  XX  XX
2  XX  XX  XX
3XX  XX  XX
4  XX  XX  XX
5XX  XX  XX
6  XX  XX  XX
7XX  XX  XX
8  XX  XX  XX
9XX  XX  XX
A  XX  XX  XX
```

ASCII VALUE = 80H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6
7
8
9
A
```

ASCII VALUE = 81H

```
 0 1 2 3 4 5 6 7
0XXXXXXXXXXXXXXX
1XXXXXXXXXXXXXXX
2XXXXXXXXXXXXXXX
3XXXXXXXXXXXXXXX
4XXXXXXXXXXXXXXX
5XXXXXXXXXXXXXXX
6XXXXXXXXXXXXXXX
7XXXXXXXXXXXXXXX
8XXXXXXXXXXXXXXX
9XXXXXXXXXXXXXXX
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 82H

```
 0 1 2 3 4 5 6 7
0XXXXXXXXXXXXXXX
1XXXXXXXXXXXXXXX
2XXXXXXXXXXXXXXX
3XXXXXXXXXXXXXXX
4XXXXXXXXXXXXXXX
5XXXXXXXXXXXXXXX
6XXXXXXXXXXXXXXX
7XXXXXXXXXXXXXXX
8XXXXXXXXXXXXXXX
9XXXXXXXXXXXXXXX
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 83H

```
 0 1 2 3 4 5 6 7
0XXXXXXXXXXXXXXX
1XXXXXXXXXXXXXXX
2XXXXXXXXXXXXXXX
3XXXXXXXXXXXXXXX
4XXXXXXXXXXXXXXX
5XXXXXXXXXXXXXXX
6XXXXXXXXXXXXXXX
7XXXXXXXXXXXXXXX
8XXXXXXXXXXXXXXX
9XXXXXXXXXXXXXXX
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 84H

```
 0 1 2 3 4 5 6 7
0XXXXXXXXXXXXXXX
1XXXXXXXXXXXXXXX
2XXXXXXXXXXXXXXX
3XXXXXXXXXXXXXXX
4XXXXXXXXXXXXXXX
5XXXXXXXXXXXXXXX
6XXXXXXXXXXXXXXX
7XXXXXXXXXXXXXXX
8XXXXXXXXXXXXXXX
9XXXXXXXXXXXXXXX
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 85H

```
 0 1 2 3 4 5 6 7
0XXXXXXXXXXXXXXX
1XXXXXXXXXXXXXXX
2XXXXXXXXXXXXXXX
3XXXXXXXXXXXXXXX
4XXXXXXXXXXXXXXX
5XXXXXXXXXXXXXXX
6XXXXXXXXXXXXXXX
7XXXXXXXXXXXXXXX
8XXXXXXXXXXXXXXX
9XXXXXXXXXXXXXXX
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 86H

```
 0 1 2 3 4 5 6 7
0  XXXXXXXXX
1  XX      XX
2    XX
3      XX
4        XX
5      XX
6    XX
7  XX      XX
8  XXXXXXXXX
9
A
```

ASCII VALUE = 87H

```
 0 1 2 3 4 5 6 7
0        XXXX
1      XX
2      XX
3      XX
4      XX
5      XX
6      XX
7      XX
8      XX
9  XXXX
A
```

ASCII VALUE = 88H

```
 0 1 2 3 4 5 6 7
0      XXXXXX
1    XX
2    XX
3    XX
4    XX
5XX  XX
6XX  XX
7  XXXX
8  XXXX
9    XX
A
```

ASCII VALUE = 89H

```
 0 1 2 3 4 5 6 7
0    XXXXXX
1  XX
2    XXXX
3  XX    XX
4    XX    XX
5      XXXX
6          XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 8AH

```
 0 1 2 3 4 5 6 7
0
1
2
3  XX    XX
4  XX    XX
5  XX    XX
6  XX    XX
7  XXXXXX  XX
8  XX
9XX
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 8BH

```
 0 1 2 3 4 5 6 7
0  XX      XX
1      XX
2    XX  XX
3  XX      XX
4  XX      XX
5  XXXXXXXXXX
6  XX      XX
7  XX      XX
8
9
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 8CH

```
 0 1 2 3 4 5 6 7
0  XX      XX
1
2    XXXXXX
3  XX      XX
4  XX      XX
5  XX      XX
6  XX      XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 8DH

```
 0 1 2 3 4 5 6 7
0  XX      XX
1
2  XX      XX
3  XX      XX
4  XX      XX
5  XX      XX
6  XX      XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 8EH

```
 0 1 2 3 4 5 6 7
0    XXXX
1  XX    XX
2  XX    XX
3    XXXX
4
5
6
7
8
9
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 8FH

```
 0 1 2 3 4 5 6 7
0    XXXX
1  XX    XX
2        XX
3    XXXX
4  XX
5  XXXXXXX
6
7
8
9
AXXXXXXXXXXXXXXX
```

ASCII VALUE = 90H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4XXXXXXXXXXXXXXXX
5
6XXXXXXXXXXXXXXXX
7
8
9
A
```

ASCII VALUE = 91H

```
 0 1 2 3 4 5 6 7
0 XX   XX
1 XX   XX
2 XX   XX
3 XX   XX
4 XX   XX
5 XX   XX
6 XX   XX
7 XX   XX
8 XX   XX
9 XX   XX
A XX   XX
```

ASCII VALUE = 92H

```
 0 1 2 3 4 5 6 7
0 XX   XX
1 XX   XX
2 XX   XX
3 XX   XX
4XXXX  XXXX
5
6XXXX  XXXX
7 XX   XX
8 XX   XX
9 XX   XX
A XX   XX
```

ASCII VALUE = 93H

```
 0 1 2 3 4 5 6 7
0 XX   XX
1 XX   XX
2 XX   XX
3 XX   XX
4 XX   XXXX
5 XX
6 XXXXXXXXXX
7
8
9
A
```

ASCII VALUE = 94H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4XXXXXXXXXX
5      XX
6XXXX  XX
7 XX   XX
8 XX   XX
9 XX   XX
A XX   XX
```

ASCII VALUE = 95H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4 XXXXXXXXXX
5 XX
6 XX   XXXX
7 XX   XX
8 XX   XX
9 XX   XX
A XX   XX
```

ASCII VALUE = 96H

```
 0 1 2 3 4 5 6 7
0 XX   XX
1 XX   XX
2 XX   XX
3 XX   XX
4XXXX  XX
5      XX
6XXXXXXXXX
7
8
9
A
```

ASCII VALUE = 97H

```
 0 1 2 3 4 5 6 7
0 XX   XX
1 XX   XX
2 XX   XX
3 XX   XX
4 XX   XXXX
5 XX
6 XX   XXXX
7 XX   XX
8 XX   XX
9 XX   XX
A XX   XX
```

ASCII VALUE = 98H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4XXXXXXXXXXXXXXXX
5
6XXXX  XXXXXXXX
7 XX   XX
8 XX   XX
9 XX   XX
A XX   XX
```

ASCII VALUE = 99H

```
 0 1 2 3 4 5 6 7
0 XX   XX
1 XX   XX
2 XX   XX
3 XX   XX
4XXXX  XX
5      XX
6XXXX  XX
7 XX   XX
8 XX   XX
9 XX   XX
A XX   XX
```

ASCII VALUE = 9AH

```
 0 1 2 3 4 5 6 7
0 XX   XX
1 XX   XX
2 XX   XX
3 XX   XX
4XXXX  XXXXXXXX
5
6XXXXXXXXXXXXXXXX
7
8
9
A
```

ASCII VALUE = 9BH

```
 0 1 2 3 4 5 6 7
0
1 XX   XX
2
3   XXXXXX
4 XX   XX
5 XX   XX
6 XX   XX
7   XXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = 9CH

```
 0 1 2 3 4 5 6 7
0
1 XX   XX
2
3   XXXX
4 XX   XX
5 XX   XX
6 XX   XX
7   XXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = 9DH

```
 0 1 2 3 4 5 6 7
0
1 XX   XX
2
3 XX   XX
4 XX   XX
5 XX   XX
6 XX   XX
7   XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = 9EH

```
 0 1 2 3 4 5 6 7
0
1   XXXX
2 XX   XX
3 XX   XX
4 XX XX
5 XX   XX
6 XX   XX
7 XX XX
8 XX
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = 9FH

```
 0 1 2 3 4 5 6 7
0   XXXX
1 XX   XX
2     XX
3       XX
4 XX   XX
5   XXXX
6
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A0H

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A1H

```
 0 1 2 3 4 5 6 7
0
1     XX
2     XX
3     XX
4     XX
5     XX
6
7     XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A2H

```
 0 1 2 3 4 5 6 7
0
1   XX XX
2   XX XX
3
4
5
6
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A3H

```
 0 1 2 3 4 5 6 7
0
1   XX XX
2   XX XX
3 XXXXXXXXX
4   XX XX
5 XXXXXXXXX
6   XX XX
7   XX XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A4H

```
 0 1 2 3 4 5 6 7
0
1     XX
2   XXXXXXX
3 XX XX
4   XXXXX
5     XX XX
6 XXXXXXX
7     XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A5H

```
 0 1 2 3 4 5 6 7
0
1 XXXX
2 XXXX   XX
3       XX
4     XX
5   XX
6 XX   XXXX
7       XXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A6H

```
 0 1 2 3 4 5 6 7
0
1   XX
2 XX XX
3 XX XX
4   XX
5 XX XX XX
6 XX   XX
7   XXXX XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A7H

```
 0 1 2 3 4 5 6 7
0
1       XX
2     XX
3
4
5
6
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A8H

```
 0 1 2 3 4 5 6 7
0
1         XX
2       XX
3     XX
4     XX
5     XX
6       XX
7         XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = A9H

```
 0 1 2 3 4 5 6 7
0
1 XX
2   XX
3     XX
4     XX
5     XX
6   XX
7 XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = AAH

```
 0 1 2 3 4 5 6 7
0
1
2     XX
3 XX  XX  XX
4   XXXXXX
5 XX  XX  XX
6     XX
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = ABH

```
 0 1 2 3 4 5 6 7
0
1
2     XX
3     XX
4 XXXXXXXXXX
5     XX
6     XX
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = ACH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6   XXXX
7   XXXX
8     XX
9   XX
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = ADH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4 XXXXXXXXXX
5
6
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = AEH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6   XXXX
7   XXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = AFH

```
 0 1 2 3 4 5 6 7
0
1
2         XX
3       XX
4     XX
5   XX
6 XX
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B0H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXX
2 XX     XX
3 XX   XXXX
4 XX  XX  XX
5 XXXX   XX
6 XX     XX
7   XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B1H

```
 0 1 2 3 4 5 6 7
0
1     XX
2   XXXX
3     XX
4     XX
5     XX
6     XX
7   XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B2H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXX
2 XX     XX
3        XX
4      XX
5   XXXX
6 XX
7 XXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B3H

```
 0 1 2 3 4 5 6 7
0
1 XXXXXXXXXX
2         XX
3       XX
4     XXXX
5         XX
6 XX     XX
7   XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B4H

```
 0 1 2 3 4 5 6 7
0
1       XX
2     XX
3   XX
4 XX   XX
5 XXXXXXXXXX
6       XX
7       XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B5H

```
 0 1 2 3 4 5 6 7
0
1 XXXXXXXXXX
2 XX
3 XXXXXXX
4         XX
5         XX
6 XX      XX
7   XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B6H

```
 0 1 2 3 4 5 6 7
0
1     XXXX
2   XX
3 XX
4 XXXXXXX
5 XX      XX
6 XX      XX
7   XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B7H

```
 0 1 2 3 4 5 6 7
0
1 XXXXXXXXXX
2         XX
3       XX
4     XX
5   XX
6   XX
7   XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B8H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXX
2 XX     XX
3 XX     XX
4   XXXXXX
5 XX     XX
6 XX     XX
7   XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = B9H

```
 0 1 2 3 4 5 6 7
0
1   XXXXXX
2 XX     XX
3 XX     XX
4   XXXXXXX
5         XX
6       XX
7   XXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = BAH

```
 0 1 2 3 4 5 6 7
0
1
2
3   XXXX
4   XXXX
5
6   XXXX
7   XXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = BBH

```
 0 1 2 3 4 5 6 7
0
1
2
3   XXXX
4   XXXX
5
6   XXXX
7   XXXX
8     XX
9   XX
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = BCH

```
 0 1 2 3 4 5 6 7
0
1         XX
2       XX
3     XX
4   XX
5     XX
6       XX
7         XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = BDH

```
 0 1 2 3 4 5 6 7
0
1
2
3 XXXXXXXXXX
4
5 XXXXXXXXXX
6
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = BEH

```
 0 1 2 3 4 5 6 7
0
1 XX
2   XX
3     XX
4       XX
5     XX
6   XX
7 XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = BFH

```
 0 1 2 3 4 5 6 7
0
1   XXXXXX
2 XX     XX
3       XX
4     XX
5     XX
6
7     XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C0H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX  XXXXXX
4  XX  XX  XX
5  XX  XXXXXX
6  XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C1H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX      XX
4  XX      XX
5  XXXXXXXXXX
6  XX      XX
7  XX      XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C2H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXX
2  XX      XX
3  XX      XX
4  XXXXXXXX
5  XX      XX
6  XX      XX
7  XXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C3H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX
4  XX
5  XX
6  XX      XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C4H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXX
2  XX      XX
3  XX      XX
4  XX      XX
5  XX      XX
6  XX      XX
7  XXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C5H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2  XX
3  XX
4  XXXXXXXX
5  XX
6  XX
7  XXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C6H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2  XX
3  XX
4  XXXXXXXX
5  XX
6  XX
7  XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C7H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX
4  XX  XXXXXX
5  XX      XX
6  XX      XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C8H

```
 0 1 2 3 4 5 6 7
0
1  XX      XX
2  XX      XX
3  XX      XX
4  XXXXXXXXXX
5  XX      XX
6  XX      XX
7  XX      XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = C9H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2      XX
3      XX
4      XX
5      XX
6      XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = CAH

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXX
2        XX
3        XX
4        XX
5        XX
6  XX    XX
7    XXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = CBH

```
 0 1 2 3 4 5 6 7
0
1  XX      XX
2  XX    XX
3  XX  XX
4  XXXX
5  XX  XX
6  XX    XX
7  XX      XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = CCH

```
 0 1 2 3 4 5 6 7
0
1  XX
2  XX
3  XX
4  XX
5  XX
6  XX
7  XXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = CDH

```
 0 1 2 3 4 5 6 7
0
1  XX      XX
2  XXXX  XXXX
3  XX  XX  XX
4  XX  XX  XX
5  XX      XX
6  XX      XX
7  XX      XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = CEH

```
 0 1 2 3 4 5 6 7
0
1  XX      XX
2  XXXX    XX
3  XX  XX  XX
4  XX    XXXX
5  XX      XX
6  XX      XX
7  XX      XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = CFH

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX      XX
4  XX      XX
5  XX      XX
6  XX      XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D0H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXX
2  XX      XX
3  XX      XX
4  XXXXXXXX
5  XX
6  XX
7  XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D1H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX      XX
4  XX      XX
5  XX  XX  XX
6  XX    XX
7    XXXX  XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D2H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXX
2  XX      XX
3  XX      XX
4  XXXXXXXX
5  XX  XX
6  XX    XX
7  XX      XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D3H

```
 0 1 2 3 4 5 6 7
0
1    XXXXXX
2  XX      XX
3  XX
4    XXXXXX
5          XX
6  XX      XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D4H

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2      XX
3      XX
4      XX
5      XX
6      XX
7      XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D5H

```
 0 1 2 3 4 5 6 7
0
1  XX      XX
2  XX      XX
3  XX      XX
4  XX      XX
5  XX      XX
6  XX      XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D6H

```
 0 1 2 3 4 5 6 7
0
1  XX      XX
2  XX      XX
3  XX      XX
4    XX  XX
5    XX  XX
6      XX
7      XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D7H

```
 0 1 2 3 4 5 6 7
0
1  XX      XX
2  XX      XX
3  XX      XX
4  XX  XX  XX
5  XX  XX  XX
6  XX  XX  XX
7    XX  XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D8H

```
 0 1 2 3 4 5 6 7
0
1  XX       XX
2  XX       XX
3    XX   XX
4       XX
5    XX   XX
6  XX       XX
7  XX       XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = D9H

```
 0 1 2 3 4 5 6 7
0
1  XX       XX
2  XX       XX
3    XX   XX
4       XX
5       XX
6       XX
7       XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = DAH

```
 0 1 2 3 4 5 6 7
0
1  XXXXXXXXXX
2           XX
3         XX
4       XX
5     XX
6  XX
7  XXXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = DBH

```
 0 1 2 3 4 5 6 7
0       XXXXXX
1       XX
2       XX
3       XX
4       XX
5       XX
6       XX
7       XX
8       XXXXXX
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = DCH

```
 0 1 2 3 4 5 6 7
0
1
2  XX
3    XX
4      XX
5        XX
6          XX
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = DDH

```
 0 1 2 3 4 5 6 7
0  XXXXXX
1       XX
2       XX
3       XX
4       XX
5       XX
6       XX
7       XX
8  XXXXXX
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = DEH

```
 0 1 2 3 4 5 6 7
0
1       XX
2     XXXXXX
3  XXXXXXXXXX
4       XX
5       XX
6       XX
7       XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = DFH

```
 0 1 2 3 4 5 6 7
0
1
2
3
4
5
6
7
8
9XXXXXXXXXXXXXXXX
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E0H

```
 0 1 2 3 4 5 6 7
0
1    XX
2      XX
3        XX
4
5
6
7
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E1H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXXXX
4  XX     XX
5  XX     XX
6  XX     XX
7    XXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E2H

```
 0 1 2 3 4 5 6 7
0
1  XX
2  XX
3  XXXXXX
4  XX     XX
5  XX     XX
6  XX     XX
7  XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E3H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXX
4  XX     XX
5  XX
6  XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E4H

```
 0 1 2 3 4 5 6 7
0
1        XX
2      . XX
3    XXXXXX
4  XX     XX
5  XX     XX
6  XX     XX
7    XXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E5H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXX
4  XX     XX
5  XXXXXX
6  XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E6H

```
 0 1 2 3 4 5 6 7
0
1      XXXX
2    XX
3    XX
4  XXXXXX
5    XX
6    XX
7    XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E7H

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXXXX
4  XX     XX
5  XX     XX
6  XX     XX
7    XXXXXX
8        XX
9  XXXXXX
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E8H

```
 0 1 2 3 4 5 6 7
0
1  XX
2  XX
3  XXXXXX
4  XX     XX
5  XX     XX
6  XX     XX
7  XX     XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = E9H

```
 0 1 2 3 4 5 6 7
0
1      XX
2
3    XXXX
4      XX
5      XX
6      XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = EAH

```
 0 1 2 3 4 5 6 7
0
1        XX
2
3      XXXX
4        XX
5        XX
6        XX
7        XX
8        XX
9    XXXX
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = EBH

```
 0 1 2 3 4 5 6 7
0
1  XX
2  XX
3  XX     XX
4  XX   XX
5  XXXX
6  XX   XX
7  XX     XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = ECH

```
 0 1 2 3 4 5 6 7
0
1    XXXX
2      XX
3      XX
4      XX
5      XX
6      XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = EDH

```
 0 1 2 3 4 5 6 7
0
1
2
3  XXXX  XX
4  XX  XX  XX
5  XX  XX  XX
6  XX  XX  XX
7  XX  XX  XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = EEH

```
 0 1 2 3 4 5 6 7
0
1
2
3  XXXXXX
4  XX     XX
5  XX     XX
6  XX     XX
7  XX     XX
8
9
AXXXXXXXXXXXXXXXX
```

ASCII VALUE = EFH

```
 0 1 2 3 4 5 6 7
0
1
2
3    XXXX
4  XX     XX
5  XX     XX
6  XX     XX
7    XXXX
8
9
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = F0H
 0 1 2 3 4 5 6 7
0
1
2
3  XXXXXX
4  XX    XX
5  XX    XX
6  XX    XX
7  XXXXXX
8  XX
9  XX
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = F1H
 0 1 2 3 4 5 6 7
0
1
2
3    XXXXXX
4  XX    XX
5  XX    XX
6  XX    XX
7    XXXXXX
8        XX
9        XX
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = F2H
 0 1 2 3 4 5 6 7
0
1
2
3  XX  XXXX
4  XXXX
5  XX
6  XX
7  XX
8
9
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = F3H
 0 1 2 3 4 5 6 7
0
1
2
3    XXXXXX
4  XX
5    XXXX
6        XX
7  XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = F4H
 0 1 2 3 4 5 6 7
0
1      XX
2      XX
3    XXXXXX
4      XX
5      XX
6      XX
7      XXXX
8
9
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = F5H
 0 1 2 3 4 5 6 7
0
1
2
3  XX    XX
4  XX    XX
5  XX    XX
6  XX    XX
7    XXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = F6H
 0 1 2 3 4 5 6 7
0
1
2
3  XX      XX
4  XX      XX
5    XX  XX
6    XX  XX
7      XX
8
9
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = F7H
 0 1 2 3 4 5 6 7
0
1
2
3  XX      XX
4  XX      XX
5  XX  XX  XX
6  XX  XX  XX
7    XX  XX
8
9
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = F8H
 0 1 2 3 4 5 6 7
0
1
2
3  XX      XX
4    XX  XX
5      XX
6    XX  XX
7  XX      XX
8
9
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = F9H
 0 1 2 3 4 5 6 7
0
1
2
3  XX      XX
4  XX      XX
5  XX      XX
6    XX  XX
7      XX
8    XX
9  XX
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = FAH
 0 1 2 3 4 5 6 7
0
1
2
3  XXXXXXXXXX
4        XX
5      XX
6    XX
7  XXXXXXXXX
8
9
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = FBH
 0 1 2 3 4 5 6 7
0        XXXX
1        XX
2      XX
3      XX
4    XX
5      XX
6      XX
7        XX
8        XXXX
9
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = FCH
 0 1 2 3 4 5 6 7
0      XX
1      XX
2      XX
3      XX
4      XX
5      XX
6      XX
7      XX
8      XX
9
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = FDH
 0 1 2 3 4 5 6 7
0  XXXX
1    XX
2      XX
3      XX
4        XX
5      XX
6      XX
7    XX
8  XXXX
9
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = FEH
 0 1 2 3 4 5 6 7
0
1    XX
2  XX  XX  XX
3        XX
4
5
6
7
8
9
AXXXXXXXXXXXXXXXX
```

```
ASCII VALUE = FFH
 0 1 2 3 4 5 6 7
0  XX  XX  XX
1XX  XX  XX
2  XX  XX  XX
3XX  XX  XX
4  XX  XX  XX
5XX  XX  XX
6  XX  XX  XX
7XX  XX  XX
8  XX  XX  XX
9XX  XX  XX
AXXXXXXXXXXXXXXXX
```