

Paperware 3

Universelles Terminal

Paperware 3 enthält ergänzende Informationen zum Universellen Terminal. Dieses Terminal verwendet die VDU-Karte von Elektor, die im September 1983 veröffentlicht wurde. Als CRT-Controller wird ein 6845 von Motorola oder ein 6545 von Rockwell/Synertek verwendet. Mit diesem IC kann auf einfache Weise ein leistungsfähiges und nachbaufreundliches Video-Interface aufgebaut werden. Folgende Kapitel sind in dieser Broschüre enthalten:

- * Informationen über die Hardware des 6845/6545
- * Programmierung und Registerstruktur des 6845/6545
- * Beschreibung des ACIA 6551 und dessen Programmierung
- * Beschreibung der Schnittstellen:
Centronics, parallel, V 24/RS 232
- * Source-Listing der Video-Routinen für das Universelle Terminal



Mit dem CRT-Controller 6845/6545 lassen sich auf einfache Weise alle Computer an einen Video-Monitor anschließen. Da der CRT-Controller programmierbar ist, können diverse Bildschirmformate durch Software erzeugt werden, ohne daß die Hardware des Video-Interfaces geändert werden muß. Wir haben mit dem 6845/6545-CRT-Controller ein Video-Interface entworfen, da dieser CRT-Controller billig ist, und nur ein paar TTL-ICs nötig sind, um auf einen Bildschirm schreiben zu können.

Bild 1 zeigt die Pinbelegung und die interne Struktur des CRT-Controllers. Zählerschaltungen für die horizontale und die vertikale Aufbereitung des Bildschirms, ein linearer Adreßzähler, sowie eine Register-File (= eine Anordnung von mehreren programmierbaren Registern) sind in einem IC vereint. Die Zählerschaltungen erzeugen auch die Steuersignale für den Video-Monitor: HSYNC, VSYNC, Display Enable und die Rastersignale RA0...RA4. Die Adreßleitungen MA0...MA13 können einen Bildschirm-speicher bis zu 16 KBytes adressieren. Da der System-Prozessor, an den der CRT-Controller angeschlossen ist, die Register-File lesen oder mit Daten beschreiben kann, läßt sich der CRT-Controller programmieren. Man kann auch festlegen, wieviele Buchstaben eine Buchstabenreihe enthalten soll, wieviele (Video-)Zeilen eine Buchstabenreihe überstreichen, wieviele Buchstabenreihen der Bildschirm enthalten soll, oder wie breit der obere, untere, linke oder rechte Rand des Bildschirms sein soll. Es läßt sich außerdem programmieren, welche Abmessungen der Cursor auf dem Bildschirm haben soll, ob der Cursor blinken oder nicht blinken soll. Sogar die Blinkfrequenz des Cursors kann von Software gesteuert werden. Da in den verschiedenen Ländern die horizontale und die vertikale Videofrequenz bei den Monitoren stark voneinander abweichen, lassen sich auch die Abstände der horizontalen und der vertikalen Synchronimpulse programmieren.

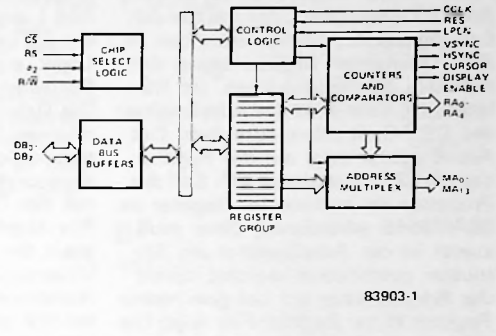
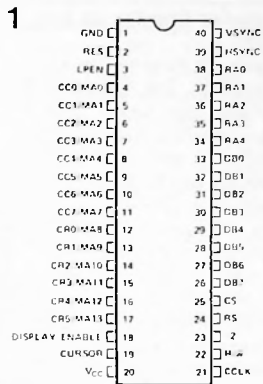
Beschreibung der Pins des 6845/6545

Pin 1 und Pin 20

An diese beiden Pins wird die Betriebsspannung für den CRT-Controller angelegt. Pin 1 ist Masse, und an Pin 20 liegt die Betriebsspannung von +5 Volt ($\pm 10\%$).

Pin 2 (RES)

Pin 2 ist der RESET-Eingang des CRT-Controllers. Dieser Eingang ist bei logisch 0 aktiv. Ist dieser Pin aktiv, dann wird auf die Adreßleitungen MA0...MA13 der Inhalt der Register R12/R14 (Display-Start-Adresse) aus-



gegeben. Auch die internen Zähler werden zurückgesetzt, und alle übrigen Ausgänge gehen nach logisch 0. Keines der internen Register wird durch die RESET-Leitung beeinflusst.

Pin 3 (LPEN)

Pin 3 ist der Lichtgriffel-Eingang des CRT-Controllers. Wird der Lichtgriffel auf einen Buchstaben auf dem Datensichtgerät gehalten, dann gibt er einen Impuls ab, wenn der Elektronenstrahl des Monitors den Buchstaben überstreicht. Der Lichtgriffel-Eingang reagiert auf einen positiven Impuls. Die Adresse des Buchstabens auf dem Monitor wird dann ins Lichtgriffel-Register kopiert. Der Prozessor kann anschließend die Adresse lesen, auf die der Lichtgriffel zeigt.

Pin 4... Pin 17 (MA0...MA13)

MA0...MA13 sind die Adreßleitungen für das Video-RAM. Diese Ausgänge werden verwendet, um die Zeichen im Video-RAM nacheinander abzufragen. Die momentane Adresse auf den Adreßleitungen bestimmt, welcher Buchstabe im Video-RAM gerade auf dem Bildschirm dargestellt wird.

Pin 18 (Display Enable)

Das Bild, das der CRT-Controller auf dem Datensichtgerät erzeugt, besteht aus einem aktiven und einem inaktiven Anteil. Das Schirmbild besteht zu einem Teil aus Buchstabenzeilen und zum anderen aus buchstabenlosen Bildrändern. Überstreicht der Elektronenstrahl des Monitors eine Buchstabenreihe, dann adressiert der CRT-Controller den aktiven Teil des Bildschirms; überstreicht der Elektronenstrahl des Monitors den Rand des Videobildes, dann adressiert der CRT-Controller den inaktiven Teil des Bildschirms. Ist der Display-Enable-Ausgang des 6845/6545 logisch 1, dann adressiert der CRT-Controller den aktiven Teil des Videobildes. Bei logisch 0 wird der inaktive Teil des Bildschirms adressiert. Das Display-Enable-Signal wird zur Dunkel-tastung des Videosignales und zur

Steuerung der peripheren Logik des CRT-Controllers verwendet.

Pin 19 (Cursor)

Pin 19 ist ein Ausgang und ist logisch 1, wenn der CRT-Controller die Stelle auf dem Bildschirm adressiert, an der sich der Cursor befindet. Die Position des Cursors auf dem Bildschirm ist frei programmierbar. Der Inhalt des Cursor-Registers in der Register-File bestimmt die Position des Cursors auf dem Bildschirm. Der Cursorausgang wird der peripheren Logik für die Cursorsteuerung zugeführt.

Pin 21 (CCLK)

Pin 21 ist der "Character Clock Input" (CCLK) des CRT-Controllers. Die Frequenz an diesem Eingang bestimmt die Buchstabenfrequenz auf dem Video-Monitor. Alle internen Zähler des CRT-Controllers verwenden das CCLK-Signal als Zeitbasis. Die Pünktchenfrequenz (Dot-Frequenz) des Buchstabens ist 8mal höher als die Buchstabenfrequenz. Bei der VDU-Karte von Elektor ist die Buchstabenfrequenz von der Pünktchenfrequenz abgeleitet, wodurch das Videobild scharf und wackelfrei ist.

Pin 22 (R/W)

Der R/W-Pin des CRT-Controllers ist ein Eingang und wird von der R/W-Leitung des Prozessors gesteuert. Ein High-Signal auf dieser Leitung ermöglicht dem Prozessor, eines der internen Register des CRT-Controllers zu lesen. Ein LOW-Signal auf dieser Leitung bewirkt, daß der Prozessor Daten in eines der internen Register schreiben kann.

Pin 23 ($\Phi 2$ oder E)

Dieser Pin ist ein Eingang und wird direkt vom Datentakt des Prozessors angesteuert. Da der Z80-Prozessor eine asynchrone Busstruktur hat, wird dieser Pin bei der VDU-Karte von Elektor über eine auf der Platine vorhandene Schnittstelle angesteuert. Alle Datenübertragungen zu oder aus dem CRT-Controller werden über dieses Signal gesteuert.

Pin 24 (RS)

Pin 24 ist der Register-Select-Eingang des CRT-Controllers, der mit der A0-Leitung des Prozessors verbunden ist. Alle 17 internen Register lassen sich mit dem RS-Pin anwählen. Ist RS logisch 0, dann wird das Adreßregister des CRT-Controllers adressiert. Das Adreßregister zeigt auf das Register, das die CPU adressieren soll. Soll der Prozessor ein bestimmtes Register im 6845/6545 adressieren, dann muß zuerst in das Adreßregister ein Bitmuster geschrieben werden, damit das Adreßregister auf das gewünschte Register in der Register-File zeigt. Die RS-Leitung ist bei dieser Operation logisch 0.

Das gewünschte Register in der Register-File ist jetzt durch das Adreßregister adressiert, und der Prozessor kann das Register lesen oder beschreiben. Dazu wird die RS-Leitung logisch 1 geschaltet, und der Prozessor schreibt in das Register oder liest aus dem Register, auf das das Adreßregister im CRT-Controller zeigt.

Pin 25 (CS)

Die Chip-Select-Leitung ist ein Eingang und ist über einen Adreßdekoder mit dem Adreßbus des Prozessors verbunden. Der CRT-Controller wird durch diese Leitung angewählt.

Pin 26 ... Pin 33 (DB7 ... DB0)

Die Leitungen DB7 ... DB0 sind mit dem Datenbus des Prozessors verbunden. Die R/W-Leitung bestimmt die Datenflußrichtung auf diesen Leitungen. Ist der CRT-Controller über die CS-Leitung nicht angewählt, dann sind die Leitungen DB7 ... DB0 hochohmig.

Pin 34 ... Pin 38 (RA4 ... RA0)

Die Raster-Adreßleitungen RA4 ... RA0 sind Ausgänge, auf denen die Adressen für den "Raster Scan" stehen. Der Elektronenstrahl auf dem Video-Monitor muß mehrere Zeilen schreiben, um eine Buchstabenzeile auf dem Bildschirm abzubilden. Die Adresse auf den Raster-Adreßleitungen gibt darüber Auskunft, in welcher Zeile sich der Elektronenstrahl innerhalb einer Buchstabenreihe befindet. Die Anzahl der Zeilen pro Buchstabenreihe ist frei programmierbar. Daher läßt sich auch der Abstand zweier Buchstabenzeilen programmieren.

Pin 39 (HSYNC)

Pin 39 ist der horizontale Synchronausgang des CRT-Controllers. Die Breite des horizontalen Synchronimpulses und der Abstand zweier Impulse ist frei programmierbar.

Pin 40 (VSYNC)

Pin 40 ist der vertikale Synchronausgang des CRT-Controllers. Die Breite des vertikalen Synchronimpulses und

der Abstand zweier Impulse ist frei programmierbar.

Bild 1 zeigt auch die interne Struktur des CRT-Controllers. Die Chip-Select-Logik ermöglicht das Lesen und Schreiben aus/in der/die Register-File. Die Datenbus-Puffer entkoppeln den internen Datenbus des 6845 vom externen Computerbus. Die Zähler und Komparatoren erzeugen zusammen mit den Daten, die in der Register-File abgelegt sind, diverse Videosignale, die für die Aufarbeitung des Videosignales benötigt werden. Der Adreßmultiplexer erzeugt die Adressen für die Buchstaben auf dem Video-Monitor.

Für den Programmierer ist die Register-File des 6845 von besonderer Bedeutung. Diese Register enthalten alle Informationen, die für die Aufbereitung des Bildes auf dem Monitor wichtig sind: zum Beispiel die Anzahl der Buchstaben je Zeile, wieviele Buchstabenreihen auf den Videoschirm geschrieben werden sollen, wieviele Zeilen der Elektronenstrahl auf den Monitor schreiben soll, um eine Buchstabenreihe zu "überstreichen". Ist ein Lichtgriffel angeschlossen, dann kann der Prozessor durch Lesen des Lichtgriffel-Registers in der Register-File feststellen, an welcher Bildschirmadresse der Buchstabe steht, über den der Lichtgriffel gehalten wird. Diese und viele andere Informationen sind in der Register-File enthalten.

Bild 2 zeigt, wie der 6845 ein Video-bild aufbaut. Horizontal wird der zeitliche Ablauf in Buchstabenbreiten gezählt. Vertikal wird in Buchstabenreihen gezählt. Das verwendete Bildschirmformat ist 80 x 24 Buchstaben. Die Buchstaben 0 ... 79 sind in der ersten Zeile auf dem Bildschirm zu sehen. Die Buchstaben 80 ... 89 bilden den dunklen linken und rechten Bildrand auf dem Bildschirm. Die Zeitspanne, die verstreicht bis der CRT-Controller von 0 ... 89 zählt, ist auf 64 Mikrosekunden eingestellt. Die Pünktchenfrequenz ist 8mal höher als die Buchstabenfrequenz. Beide

Frequenzen sind bei der Elektor-VDU-Karte starr miteinander gekoppelt und werden von einem gemeinsamen Taktgenerator abgeleitet.

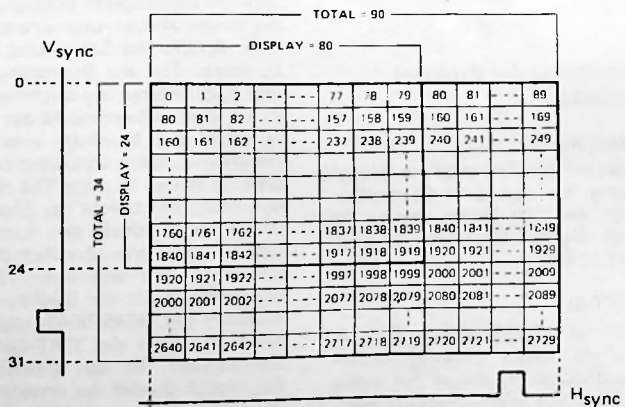
Da der horizontale Synchronimpuls in die Mitte der nicht aktiven Bildschirmbuchstaben 80 ... 89 fällt, kommt die eine Hälfte dieses Bereiches auf dem rechten Bildrand und die andere auf dem linken Bildrand zu stehen.

Eine Buchstabenreihe besteht aus mehreren Videozeilen, die der Elektronenstrahl des Video-Monitors schreiben muß, um eine Buchstabenzeile zu überstreichen. Bei der Elektor-VDU-Karte überstreichen 9 Videozeilen eine Buchstabenreihe, wobei die schwarzen Leerzeilen zwischen den Buchstabenreihen mitgezählt sind. Eine Buchstabenreihe dauert also 9 mal 64 Mikrosekunden. Vertikal wird in Buchstabenreihen gezählt. Die Buchstabenreihen 0 ... 23 gehören zum aktiven Teil des Bildes auf dem Monitor. Die Buchstabenreihen 24 ... 31 liegen außerhalb des aktiven Bildteiles und bilden den oberen und den unteren Bildrand. Der vertikale Synchronimpuls fällt in die Mitte der nicht aktiven Buchstabenreihen (= leere Buchstabenreihen). Der 6845 CRT-Controller ist frei programmierbar. Das Bildschirmformat und die Lage der Synchronimpulse lassen sich auf einfache Weise durch Einschreiben von Daten in die interne Register-File festlegen. Deshalb werden alle für den CRT-Controller erforderlichen Daten nach einem RESET aus einem Inhaltsverzeichnis gelesen und in die Register-File des Controllers kopiert. Wie man die Register-File programmieren kann wollen wir jetzt beschreiben.

Die Register-File des 6845/6545 besteht aus 21 Registern, die bis zu 8 bit breit sind. Alle Register werden über eine einzige Adreßleitung angewählt.

Ein Register in der Register-File wird angewählt, indem das Adreßregister zuerst auf das angewählte Register in der Register-File gesetzt wird. Anschließend wird die Adreßleitung des

2



83903-2

CRT-Controllers auf logisch 1 gesetzt, und der Mikroprozessor kann in das vom Adreßregister angewählte Register in der Register-File Daten schreiben. Die absoluten Adressen des Adreßregisters und der Register-File können Sie den Source-Listings am Ende dieser Paperware entnehmen.

Die Register in der Register-File:

Register R0 (horizontal total)

Dieses Register enthält die Anzahl aller aktiven und nicht aktiven Buchstaben einer Buchstabenreihe minus Eins. Die aktiven Buchstaben sind die Buchstaben, die in einer Buchstabenreihe zu sehen sind; und die nicht aktiven Buchstaben bilden den rechten und den linken Rand des Videobildes. Der Inhalt des Registers R0 bestimmt den Abstand zweier horizontaler Synchronimpulse.

Register R1 (Anzahl der Buchstaben je Buchstabenreihe)

Dieses Register enthält die Anzahl der Buchstaben je Buchstabenreihe. Sollen beispielsweise 80 Buchstaben in einer Reihe enthalten sein, dann muß \$50 (5 mal 16) in dieses Register geschrieben werden.

Register R2 (Position des horizontalen Sync.)

Dieses Register bestimmt die Lage des horizontalen Synchronimpulses innerhalb einer aktiven/nicht aktiven Buchstabenreihe. Die Position des horizontalen Synchronimpulses bestimmt die Breite des linken und des rechten Bildrandes (siehe Bild 2).

Register R3 (Breite von HSYNC und VSYNC)

Beim 6845/6545 läßt sich durch dieses Register die Breite des horizontalen Synchronimpulses programmieren. B3...b0 in Register R3 bestimmen, wieviel Buchstaben der horizontale Synchronimpuls breit ist. Der vertikale Synchronimpuls ist immer 16 Videozeilen breit. Beim SY6545 E läßt sich über b7...b4 sogar die Breite des vertikalen Synchronimpulses programmieren.

Register R4 (vertikal total)

Das Vertikal-Total-Register ist ein 7-bit-Register. Es enthält die Anzahl aller aktiven/nicht aktiven Buchstabenreihen minus Eins, die vertikal auf den Bildschirm geschrieben werden. Werden beispielsweise 32 aktive/nicht aktive Buchstabenreihen auf den Bildschirm geschrieben, dann muß das Register R4 mit \$1F = 31 geladen werden. Die Zeit, die verstreicht, bis alle 32 Buchstabenreihen geschrieben sind, ist bei der VDU-Karte von Elektor 9 mal 64 mal 32 Mikrosekunden = 18,432 Millisekunden. Da normalerweise der Abstand zweier vertikaler Synchronimpulse 20 Milli-

sekunden beträgt, ist noch eine Korrektur von 20 - 18,432 Millisekunden nötig. Diese Korrektur kann durch Hinzufügen weiterer inaktiver Buchstabenreihen (Leerzeile: 9 mal 64 Mikrosekunden = 576 Mikrosekunden) erfolgen. Die Einstellung des exakten Abstands der vertikalen Synchronimpulse auf 2 Millisekunden ist mit Register R5 in 64-µs-Schritten möglich.

Register R5 (vertikal total, Feinabstimmung)

Das Vertical-Total-Adjust-Register ist ein 5 bit breites Register. Es enthält die Anzahl der Leerzeilen, die nötig sind, um zusammen mit dem Inhalt des Registers R4 auf eine Bildwiederholfrequenz von 50 Hz (20 Millisekunden) zu kommen.

Register R6 (Anzahl der Buchstabenreihen auf dem Monitor)

Dieses 7-bit-Register enthält die Anzahl der Buchstabenreihen auf dem Bildschirm.

Register R7 (vertikale SYNC-Position)

Der Inhalt dieses Registers bestimmt die Lage des vertikalen Synchronimpulses (siehe Bild 2). Der Abstand des vertikalen Synchronimpulses wird in "Buchstabenreihen-Zeiten" gemessen. Durch die Position des vertikalen Synchronimpulses werden der obere und der untere Bildrand auf dem Bildschirm festgelegt.

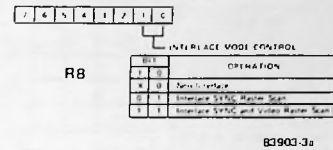
Register R8 (Betriebsart: Steuerung des CRT-Controllers)

Dieses Register ist nur 2 bit breit. Mit ihm läßt sich die Betriebsart des CRT-Controllers festlegen. Bild 3a zeigt, wie b1 und b0 dieses Registers beim 6845/6545 die gewünschte Betriebsart wählen. Bild 3c zeigt alle möglichen Betriebsarten des 6845/6545. Eine Ausnahme bildet der SY6545 E. Bei diesem Controller ist das Register R8 8 bit breit (Bild 3b). Die wichtigsten Bits in diesem erweiterten Register sind die Bits für Cursor und Display-Enable und Cursor um eine Buchstabenzeit verzögern. Bei der VDU-Karte haben wir diese Verzögerung mit externen Flipflops vorgenommen. Das hat den Vorteil, daß alle XX45-CRT-Controller auf der Elektor-Platine eingesetzt werden können!

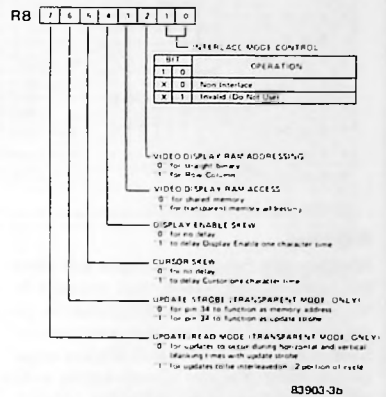
Register R9 (Scan-Line-Register)

Der Inhalt dieses Registers bestimmt, wieviele Videozeilen minus Eins über eine Buchstabenreihe auf dem Bildschirm geschrieben werden sollen. Bei der VDU-Karte werden normalerweise 9 Videozeilen über eine Buchstabenreihe geschrieben (8 aktive Zeilen und eine Video-Leerzeile).

3a

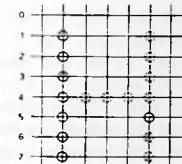


b

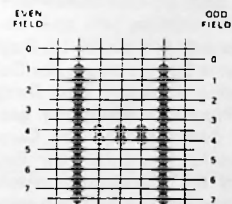


C

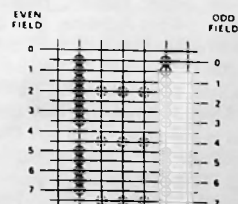
Display Modes



NON-INTERLACED



INTERLACED SYNC



INTERLACED SYNC AND VIDEO

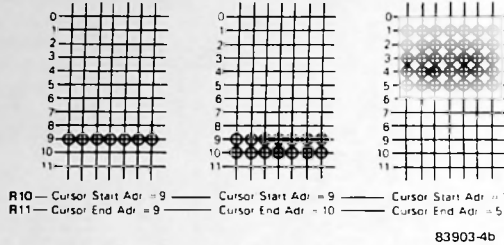
83903-3c

4a

R10		CURSOR MODE
BIT 6	BIT 5	
0	0	No Blinking
0	1	No Cursor
1	0	Blink at 16x field rate (fast)
1	1	Blink at 32x field rate (slow)

83903-4a

b



Achtung:

Werden graphische Symbole auf dem Bildschirm dargestellt, dann müssen 8 Videozeilen über eine Graphikzeile geschrieben werden, da die graphischen Symbole in einer 8-mal-8-Matrix angeordnet sind. Bei der Graphikzeile sollte die Video-Leerzeile entfallen, damit sich die Graphik-Symbole nahtlos aneinanderhängen lassen.

Register R10 und R11 (Abmessung des Cursors)

Mit diesen beiden Registern lassen sich die Abmessungen des Cursors bestimmen. Der Inhalt des Registers R10 bestimmt, in welcher Videozeile der Cursor beginnt (horizontale Abmessung), und der Inhalt des Registers R11 bestimmt, bei welcher Videozeile der Cursor zu Ende ist (vertikale Abmessung). Die beiden Bits b6 und b5 im Register R10 bestimmen, ob der Cursor mit einer bestimmten Frequenz blinken soll, ob der Cursor nicht blinken soll, oder ob kein Cursor auf dem Bildschirm sichtbar sein soll. Bild 4a zeigt, wie sich mit b5, b6 im Register R10 diverse Betriebsarten des Cursors programmieren lassen. Drei Beispiele für die Programmierung der geometrischen Abmessungen des Cursors zeigt Bild 4b. Der Inhalt der Register R10 und R11 bestimmt, wieviele Videozeilen der Cursor hoch sein soll.

Register R12 und R13 (Display Start Adresse)

Der Inhalt dieser beiden Register bildet eine 14-bit-Adresse, die den ersten Buchstaben auf dem Bildschirm (Buchstabe links oben = "Home Position") adressiert. Nur b0... b10 der Display-Start-Adresse werden bei der VDU-Karte verwendet, da das Display-RAM 2 KByte (0... 7FF) lang ist. R12 und R13 bilden zusammen den "Display Start Pointer". Über diesen Zeiger (Pointer) lässt sich der gesamte Bildschirminhalt zeilenweise oder seitenweise von unten nach oben oder von oben nach unten verschieben.

Diesen Vorgang nennt man Scroll Up und Scroll Down. Bild 5 zeigt, wie das Video-RAM aufgebaut ist. Es handelt sich um einen Speicher, der ringförmig angeordnet ist (Wrap Around Memory). Der gesamte Bildschirm wird von einem Hardware-Pointer ("Display Start Pointer") und drei Software-Pointern (FLN, CLN, LLN) gesteuert. Diese vier Pointer sind virtuelle Pointer, das heißt ihr Inhalt stimmt nicht mit der effektiven Adresse im Video-RAM überein. Die effektive Buchstabenadresse im Video-RAM berechnet sich aus:

$$\begin{aligned} &\text{effektive Buchstabenadresse} = \\ &\text{Inhalt von CLN (momentane Cursor-Reihe)} \\ &\text{plus} \\ &\text{Cursor-Spalte} \\ &\text{plus} \\ &\text{absolute Startadresse des Video-RAMs (SD000)} \end{aligned}$$

Definitionen:

FLN = "First Line Pointer" zeigt immer auf den Beginn der ersten Buchstabenzeile auf dem Bildschirm. Der Inhalt dieses Zeigers ist ein Hexzahl zwischen 00... 7FF.
CLN = "Current Line Pointer" zeigt immer auf den Beginn der Zeile, in

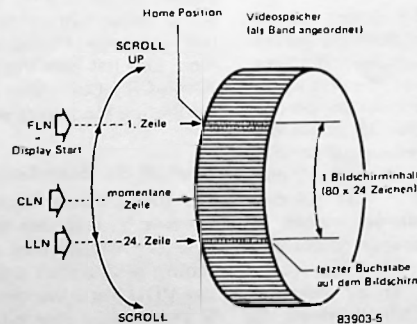
der sich der Cursor befindet. Die Cursorposition ergibt sich aus CLN plus COL = Inhalt des "Current Line Pointers" plus Spalte, in der sich der Cursor gerade befindet. Der Inhalt von CLN ist eine Hexzahl zwischen 0... 7FF.
LLN = "Last Line Pointer" zeigt immer auf den Beginn der untersten Zeile auf dem Bildschirm. Der Inhalt dieses Zeigers ist eine Hexzahl zwischen 0... 7FF.
Der Display-Start-Zeiger befindet sich im CRT-Controller und zeigt wie der FLN-Zeiger auf den ersten Buchstaben in der ersten Buchstabenzeile auf dem Bildschirm. Der Inhalt des Display-Start-Zeigers ist ebenfalls eine Hexzahl zwischen 0... 7FF.
Der Inhalt der vier Zeiger ändert sich wie folgt:

1. Inkrementiere bis maximal \$7FF. Wird dieser Wert überschritten, dann fange wieder bei Null an zu zählen.
2. Dekrementiere bis minimal \$0. Wird dieser Wert unterschritten, dann fange wieder bei \$7FF an zu zählen.

Scroll:

Bei der VDU-Karte von Elektor ist das Bildschirmformat normalerweise 80 mal 24. Der Bildschirm besteht aus einer ersten und letzten Buchstabenreihe (1. Reihe und 24. Reihe). Der Cursor auf dem Bildschirm kann in vertikaler Richtung zwischen diesen 24 Buchstabenreihen auf und ab bewegt werden. Wird der Cursor über die letzte Buchstabenreihe hinaus bewegt, dann findet ein "Scroll Up" statt. Die Video-Software bewegt in diesem Fall den gesamten Bildschirminhalt um eine Zeile nach oben und löscht die oberste Bildschirmzeile aus. Wird der Cursor über die erste Buchstabenzeile hinausbewegt, dann findet ein "Scroll Down" statt. Die Video-Software bewegt in diesem Fall den gesamten Bildschirminhalt um eine Zeile nach unten und löscht die unterste Buchstabenzeile aus.

5



Es gibt zwei Arten von "Scroll":

1. Der Speicher-Scroll. Bei diesem Verfahren wird der gesamte Inhalt des Video-RAMs um eine Zeilenlänge nach oben oder nach unten verschoben. Die Display-Start-Adresse wird nicht verändert. Bei einem Format von 80 mal 24 müssen ungefähr 1900 Buchstaben verschoben werden. Da viele Daten verschoben werden müssen, ist der Speicher-Scroll sehr langsam und prozessorintensiv. Diesen Nachteil haben wir nicht hingenommen und haben bei der VDU-Karte einen Zeiger-Scroll verwendet.

2. Der Zeiger-Scroll. Bei diesem Verfahren wird die Display-Start-Adresse im CRT-Controller modifiziert; der Inhalt des Video-RAMs wird nicht verschoben. Bei einem "Scroll Up" wird der Display-Start-Zeiger um die Anzahl der Buchstaben pro Zeile erhöht. Bei einem "Scroll Down" wird der Display-Start-Zeiger um die Anzahl der Buchstaben pro Zeile vermindert. Für einen Scroll muß nicht mehr der Inhalt des gesamten Video-RAMs nach oben oder nach unten verschoben werden, sondern die CPU muß nur zwei Bytes in der Register-File des CRT-Controllers modifizieren. Deshalb ist der Zeiger-Scroll viel schneller als der Speicher-Scroll.

6

CS	RS	Address Reg					Reg No	Register Name	Stored Info	RD	WR	Register Bit												
		4	3	2	1	0						7	6	5	4	3	2	1	0					
1	-	-	-	-	-	-	-	-	-	-	-													
0	0	-	-	-	-	-	-	Address Req	Reg. No.		✓	U	L	V										
0	0	-	-	-	-	-	-	Status Req			✓	U	L	V										
0	1	0	0	0	0	0	R0	Horiz Total - 1	# Charac.		✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	0	0	0	0	1	R1	Horiz Displayed	# Charac.		✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	0	0	0	1	0	R2	Horiz Sync Position	# Charac.		✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	0	0	0	1	1	R3	VSYNC HSYNC Widths	# Scan Lines and # Char. Times		✓	V ₃	V ₂	V ₁	V ₀	H ₃	H ₂	H ₁	H ₀					
0	1	0	0	1	0	0	R4	Vert. Total - 1	# Charac. Row		✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	0	0	1	0	1	R5	Vert. Total Adjust	# Scan Lines		✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	0	0	1	1	0	R6	Vert. Displayed	# Charac. Rows		✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	0	0	1	1	1	R7	Vert. Sync Position	# Charac. Rows		✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	0	1	0	0	0	R8	Mode Control			✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	0	1	0	0	1	R9	Scan Lines - 1	# Scan Lines		✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	0	1	0	1	0	R10	Cursor Start	Scan Line No.		✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	0	1	0	1	1	R11	Cursor End	Scan Line No.		✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	0	1	1	0	0	R12	Display Start Addr. (H)			✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	0	1	1	0	1	R13	Display Start Addr. (L)			✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	0	1	1	1	0	R14	Cursor Position (H)			✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	0	1	1	1	1	R15	Cursor Position (L)			✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	1	0	0	0	0	R16	Light Pen Req (H)			✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	1	0	0	0	1	R17	Light Pen Req (L)			✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	1	0	0	1	0	R18	Update Address Reg (H)			✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	1	0	0	1	1	R19	Update Address Reg (L)			✓	•	•	•	•	•	•	•	•	•	•	•	•	•
0	1	1	1	1	1	1	R31	Dummy Location				•	•	•	•	•	•	•	•	•	•	•	•	•

Notes: Designates binary bit
 Designates unused bit. Reading this bit is always "0", except for R31, which does not drive the data bus at all, and for CS = "1" which operates likewise.

83903-6

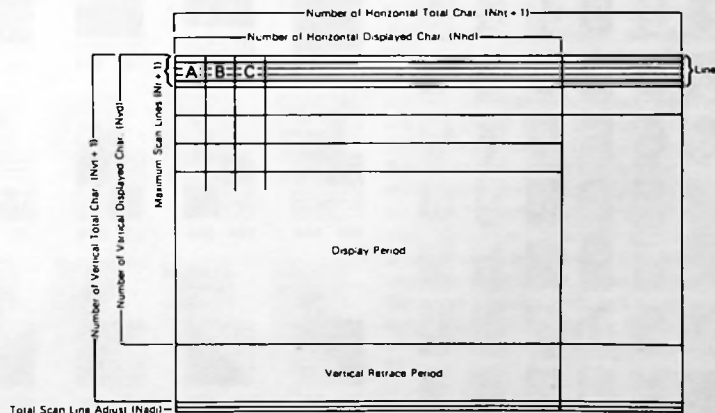
Register R14 und R15 (Cursor Position)

Der Inhalt dieser beiden Register bestimmt die Cursor-Position auf dem Bildschirm. Wenn das Bitmuster in diesen Registern gleich dem Bitmuster auf den Adreßleitungen MA0...MA10 des CRT-Controllers ist, dann wird die Cursorleitung (CUR) des CRT-Controllers aktiv und der Cursor ausgegeben. Weiter werden automatisch die RA0...RA4-Leitungen mit dem Inhalt des Cursor-Start- und Cursor-End-Registers (R10 und R11) verglichen. Dieser Vergleich entscheidet, wieviele Videozeilen der Cursor lang ist.

Register R16 und R17 (Light Pen Register)

Diese beiden Register enthalten die Adresse des Buchstabens, über den der Lichtgriffel gehalten wird. Die Adresse dieses Buchstabens wird in das Lichtgriffel-Register übernommen, wenn der Pegel am LPEN-Eingang des CRT-Controllers von logisch Null nach Eins wechselt und kurz danach eine negative Flanke der Buchstabenfrequenz entsteht. Der 6845/6545 hat noch einige andere Register, die hier nicht besprochen werden, da manche Hersteller diese Register stark verändert haben. Die Video-Software verwendet nur die beschriebenen Register. Das hat den großen Vorteil, daß alle auf dem Markt erhältlichen 6845/6545-CRT-Controller für die VDU-Karte verwendet

7



83903-7

werden können. Bild 6 gibt einen Überblick über die interne Register-File dieses CRT-Controllers. Bild 7 zeigt den gesamten Aufbau des Bildes auf dem Bildschirm. Alle verwendeten Begriffe werden nochmals kurz definiert. Für die Umsetzung eines Buchstabens in ein Punktchenmuster sorgt der Zeichengenerator. In einem EPROM (2732) ist das Punktchenmuster aller auf dem Bildschirm darstellbarer Buch-

staben und Graphikzeichen abgelegt. Da von Ohio Scientific diverse Spielprogramme erhältlich sind, haben wir die Graphikzeichen kompatibel zur Ohio-Software gemacht, allerdings einen besseren Buchstabensatz vorgesehen. Der Zeichengenerator ist in zwei 2-KByte-Hälften aufgeteilt. Die Buchstaben nehmen die Adressen 00X...7FX ein, die Graphikzeichen belegen die Adressen 80...FFX (X = 0...F).

Bild 8 zeigt den Inhalt des Zeichengenerators für die USA-Version. Die Änderungen für die deutsche Version des Zeichengenerators zeigt Bild 9.

8

1. Byte

MSD →

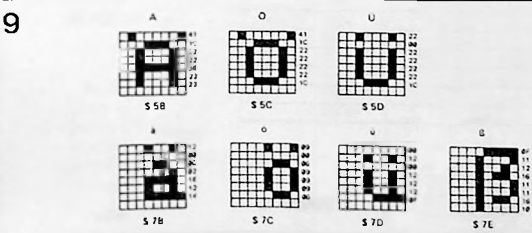
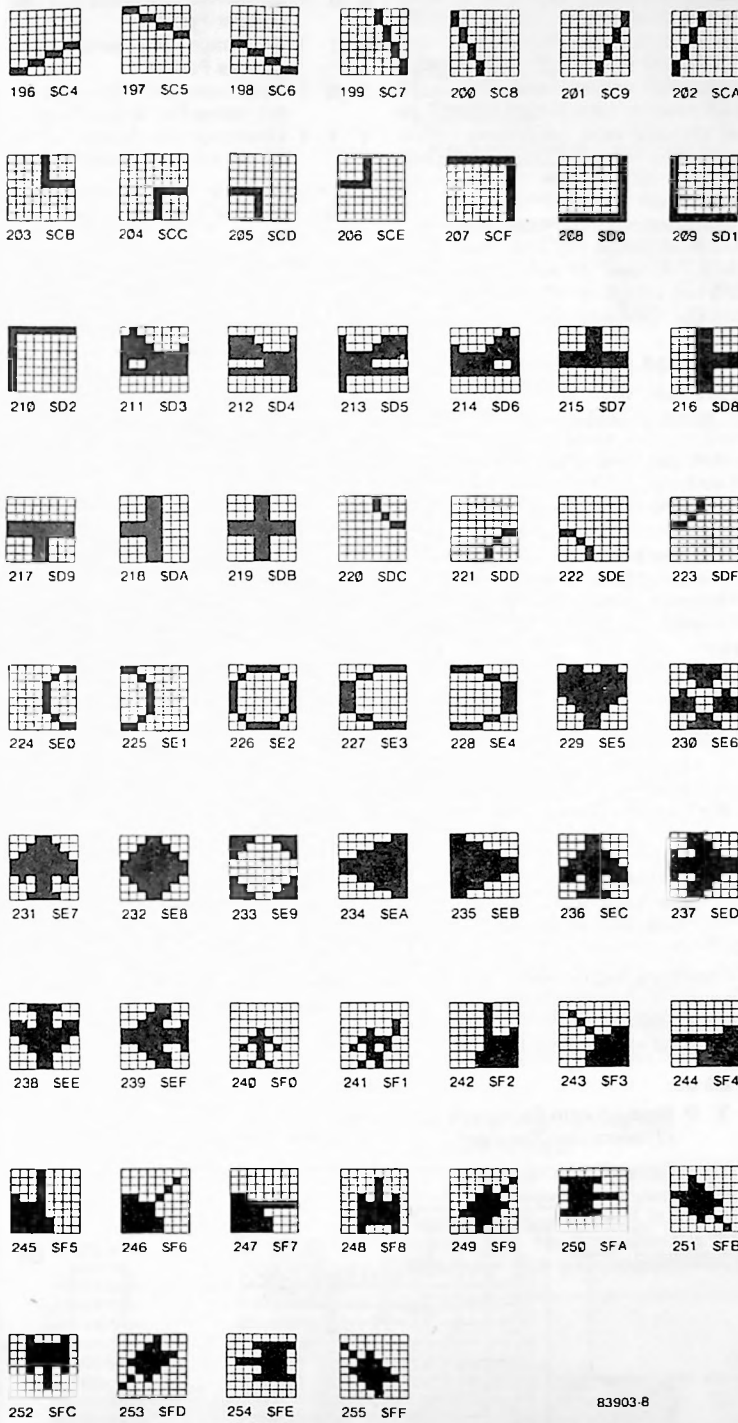
2 3 4 5 6 7

2. Byte

LSD ↓

0							
1							
2							
3							
4							
5							
6							
7							
8							
9							
A							
B							
C							
D							
E							
F							

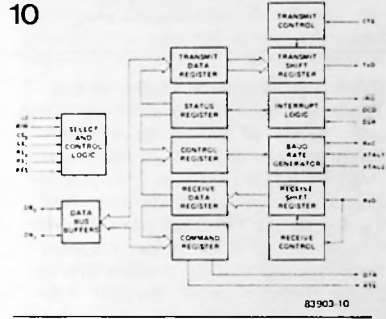
128 S80	129 S81	130 S82	131 S83	132 S84	133 S85	134 S86	135 S87	136 S88	137 S89	138 S8A	139 S8B		
140 S8C	141 S8D	142 S8E	143 S8F	144 S90	145 S91	146 S92	147 S93	148 S94	149 S95	150 S96	151 S97	152 S98	153 S99
154 S9A	155 S9B	156 S9C	157 S9D	158 S9E	159 S9F	160 SA0	161 SA1	162 SA2	163 SA3	164 SA4	165 SA5	166 SA6	167 SA7
168 S8B	169 SA9	170 SAA	171 SAB	172 SAC	173 SAD	174 SAE	175 SAF	176 SB0	177 SB1	178 SB2	179 SB3	180 SB4	181 SB5
182 S86	183 SB7	184 SB8	185 SB9	186 SBA	187 SBB	188 SBC	189 SBD	190 SBE	191 SBF	192 SC0	193 SC1	194 SC2	195 SC3



Der ACIA 6551

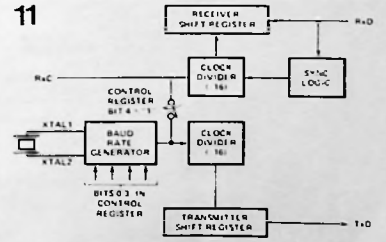
Den seriellen Datenaustausch zwischen einem Computer und dem universellen Terminal steuert ein ACIA (asynchroner Interface-Adapter) vom Typ 6551. Dieses IC hat einen internen Baudrate-Generator, zwei Steuerregister, ein Statusregister und ein Sende- und Empfangsregister. Extern müssen nur ein Quarz und ein Pegelumsetzer (TTL/V24) angeschlossen werden. Bild 10 zeigt das Blockschaltbild des 6551. Alle Handshake-Leitungen für den direkten Anschluß an ein Modem sind vorhanden.

10

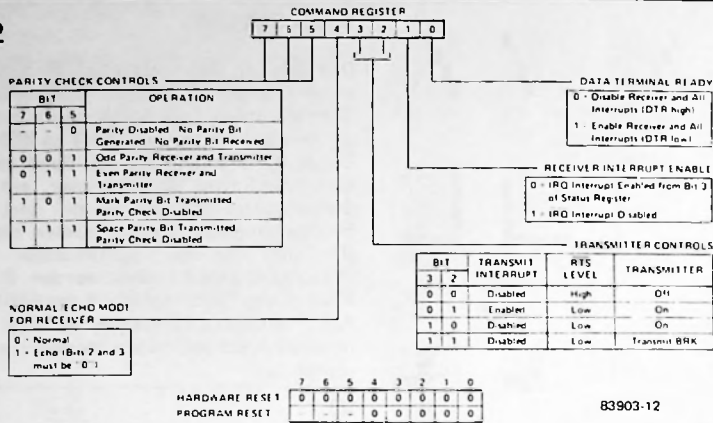


Der 6551 hat 6 interne Register, die mit den beiden Register-Select-Leitungen RS0, RS1 und der R/W-Leitung ausgewählt werden. Wie die einzelnen Register adressiert werden, zeigt Bild 11. In das Transmit-Data-Register schreibt der Prozessor den zu übertragenden Buchstaben. Mit welcher Geschwindigkeit die seriellen Daten aus dem ACIA heraus oder in den ACIA hinein übertragen werden, bestimmt der programmierbare Baudrate-Generator. Daten, die auf dem Bildschirm sichtbar sein sollen, empfängt das Terminal vom Computer durch Lesen des Receiver-Data-Registers. Der Baudrate-Generator steuert den Transmitter und den Receiver im ACIA. Es ist jedoch auch möglich, den Receiver vom Baudrate-Generator abzukoppeln und an einen externen Baudrate-Generator anzuschließen. Transmitter und Receiver können jetzt mit unterschiedlicher Geschwindigkeit arbeiten.

11



RS1	RS0	Write	Read
0	0	Transmit Data Register	Receiver Data Register
0	1	Programmed Reset (Data is "Don't Care")	Status Register
1	0	Command Register	
1	1	Control Register	



- 0 0 1 Empfange und sende mit ungerader Parität
- 0 1 1 Empfange und sende mit gerader Parität
- 1 0 1 Übertrage ein Mark-Parity-Bit; keine Paritätsprüfung
- 1 1 1 Übertrage ein Space-Parity-Bit; keine Paritätsprüfung

Bild 13 zeigt, welche Bedeutung die einzelnen Bits im Control-Register des ACIA haben:

b3 ... b0 Mit diesen Bits wird die Baudrate des ACIA programmiert. Sind alle vier Bits Null, dann arbeiten Transmitter und Receiver mit einem externen Baudrate-Generator, dessen Taktfrequenz an Pin 5 (R x C) des 6551 geführt wird. 15 verschiedene Baudraten zwischen 50 ... 19200 Baud lassen sich programmieren.

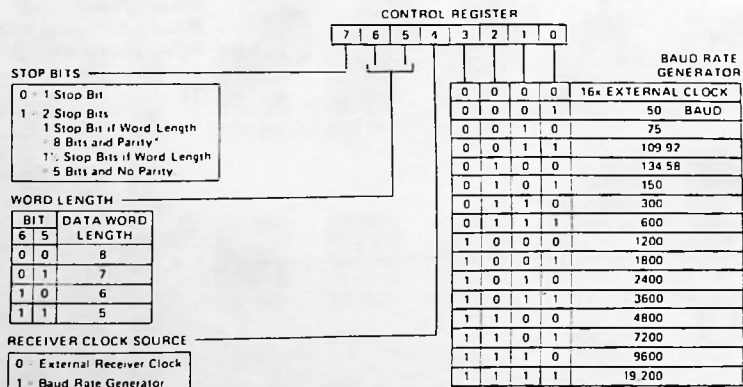
- b4:**
 - 0 = Receiver wird mit externem Baudrate-Generator betrieben. Pin 5 des 6551 ist der Takteingang.
 - 1 = Receiver und Transmitter werden vom internen Baudrate-Generator gesteuert. Pin 5 des 6551 ist der Takt- ausgang. Da der interne Baudrate-Generator weitere ACIAs ansteuern kann, spart man pro ACIA einen teuren Quarz.
- b6 b5:**
 - 0 0 Das serielle Signal hat 8 Datenbits.
 - 0 1 Das serielle Signal hat 7 Datenbits.
 - 1 0 Das serielle Signal hat 6 Datenbits.
 - 1 1 Das serielle Signal hat 5 Datenbits. Dieses Format wird bei einer Baudot-Übertragung verwendet.
- b7:**
 - Dieses Bit bestimmt die Anzahl der Stoppbits (siehe Bild 13).

Bevor der Prozessor Daten in das Transmit-Data-Register schreiben kann, oder bevor er das Receive-Data-Register lesen kann, muß das Status-Register gelesen werden. Die Bits im Status-Register geben darüber Auskunft, ob das Transmit-Data-Register bereits leer ist oder ob das Receiver-Data-Register voll ist. Andere Bits im Status-Register geben darüber Auskunft, ob bei der seriellen Übertragung ein Übertragungsfehler aufgetreten ist oder nicht ("parity error, overrun, framing error ect."/Bild 14).

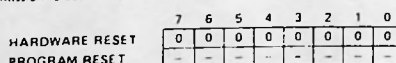
Das Command- und das Control-Register sind Lese/Schreiberegister und bestimmen die Betriebsart des ACIA. Nach einem Power-On-Reset oder manuellen RESET werden diese beiden Register von der CPU mit einem Bitmuster geladen. Die Arbeitsweise des ACIA wird somit programmiert. Bild 12 zeigt, welche Bedeutung die einzelnen Bits im Command-Register haben:

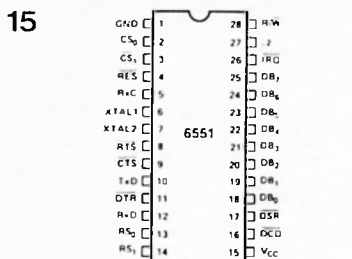
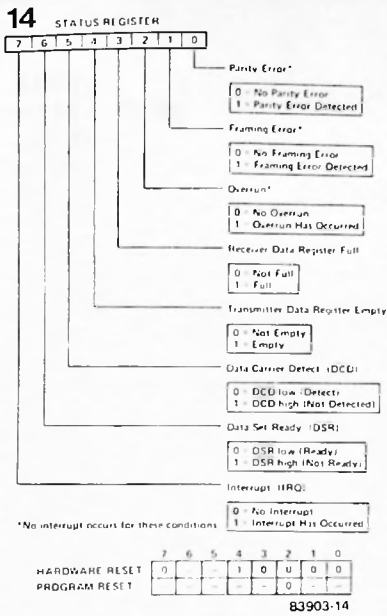
- b0:**
 - 0 = Verbiere einen Receiver-Interrupt und alle anderen Interrupts. Die Leitung DTR ist logisch 1.

- 1 = Erlaube einen Receiver-Interrupt und alle anderen Interrupts. Die Leitung DTR ist logisch 0.
- b1:**
 - 0 = Erlaube einen Interrupt, wenn das Receiver-Data-Register voll ist.
 - 1 = Verbiere einen Interrupt, wenn das Receiver-Data-Register voll ist.
- b3 b2:**
 - 0 0 = Verbiere Transmitter-Interrupt. Die Leitung RTS ist logisch Eins.
 - 0 1 = Erlaube Transmitter-Interrupt. Die Leitung RTS ist logisch Null.
 - 1 0 = Verbiere Transmitter-Interrupt. Die Leitung RTS ist logisch Null.
 - 1 1 = Verbiere Transmitter-Interrupt. Die Leitung RTS ist logisch Null. Übertrage BRK.
- b4:**
 - 0 = Normale Betriebsweise.
 - 1 = Echo-Buchstaben. Umschaltung auf Halb- und Vollduplex. B3 und b2 müssen Null sein.
- b7 b6 b5:**
 - X X 0 Erzeuge kein Paritätsbit (Transmitter/Receiver)



* This allows for 9 bit transmission (8 data bits plus parity)





MSD	0	1	2	3	4	5	6	7
LSD	000	001	010	011	100	101	110	111
0	0000	NULL	DLF	SP	0	0	0	0
1	0001	SGH	DC1	*	1	A	O	*
2	0010	STX	DC2	*	2	B	R	B
3	0011	ETX	DC3	*	3	C	S	C
4	0100	EOT	DC4	*	4	D	T	0
5	0101	ENG	NAK	*	5	E	0	*
6	0110	ACK	SYN	*	6	F	0	*
7	0111	REL	CAN	*	7	G	0	*
8	1000	BS	CAN	*	8	H	0	*
9	1001	HT	EN	*	9	I	0	*
A	1010	LF	SUB	*	10	J	0	*
B	1011	VT	ESC	*	11	K	0	*
C	1100	FF	FS	*	12	L	0	*
D	1101	CR	GS	*	13	M	0	*
E	1110	SO	RS	*	14	N	0	*
F	1111	SI	VS	*	15	O	0	DEL

83903-15

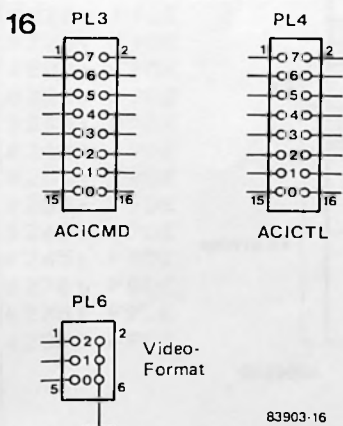


Bild 14 zeigt das Statusregister des 6551. Die am meisten verwendeten Bits in diesem Register sind die Receiver-Flags (b3) und die Transmitter-Flags (b4). Diese beiden Flags geben darüber Auskunft, ob ein neuer Buchstaben empfangen oder gesendet werden darf. Typisch Send- und Empfangsroutinen enthält das Source-Listing am Ende dieser Paperware. Der Name der ACIA-Senderroutine ist "ACIOUT" und der Name der ACIA-Empfangsroutine ist "ACIIN". Bild 15 zeigt die Anschlußbelegung des 6551 und alle übertragbaren ASCII-Zeichen.

Die Programmierung des ACIA:

Das Universelle Terminal von Elektor besteht aus der 6502-CPU-Karte und der VDU-Karte. Beide sind im Euro-kartenformat ausgeführt und können direkt auf den Elektor-Computer-Bus gesteckt werden. Da das Universelle Terminal an alle Computersysteme anzuschließen ist, ist auf der CPU-Karte zusätzliche Hardware vorgesehen, um das ACIA-Control-Register und Command-Register einfach programmieren zu können. Bild 16 zeigt, daß PL3 das ACIA-Command-Register und PL4 das ACIA-Control-Register programmiert. Die Programmierung erfolgt durch Stecken von Brücken.

Achtung! !

- 1 = gesteckte Brücke
- 0 = offene Kontakte

Die gesteckten oder nichtgesteckten Brücken auf PL3, PL4 korrespondieren mit den einzelnen Bits im ACIA-Control- und ACIA-Command-Register. Nach einem RESET liest das Terminal das gewählte Bitmuster auf PL3, PL4 und programmiert somit die Betriebsweise des ACIA. Bild 16 zeigt die Ordnung der einzelnen Bits.

Die Programmierung des Video-Formats

Durch Stecken von Brücken auf PL6 (Bild 16) lassen sich diverse Video-Formate programmieren. Beim Initialisieren liest der Prozessor das Bit-Muster auf PL6 und programmiert das

gewünschte Bildschirmformat. Acht Bildschirm-Formate sind im EPROM abgelegt, wovon eines noch für den Anwender frei ist. Folgende Formate stehen zur Verfügung:

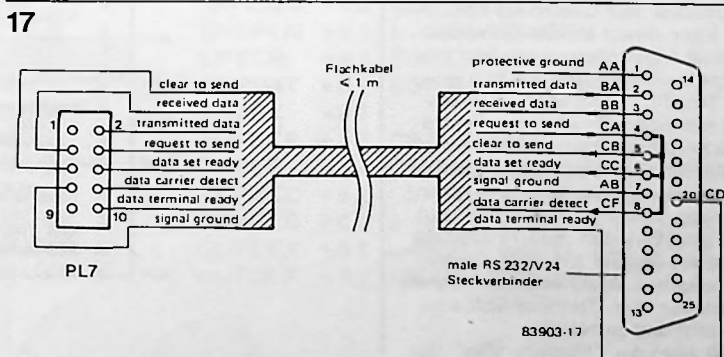
- 0 0 0 80 mal 24
- 0 0 1 80 mal 25
- 0 1 0 64 mal 16
- 0 1 1 64 mal 24
- 1 0 0 90 mal 22
- 1 0 1 48 mal 12
- 1 1 0 24 mal 24
- 1 1 1 frei programmierbar

RS232/V24 Interface

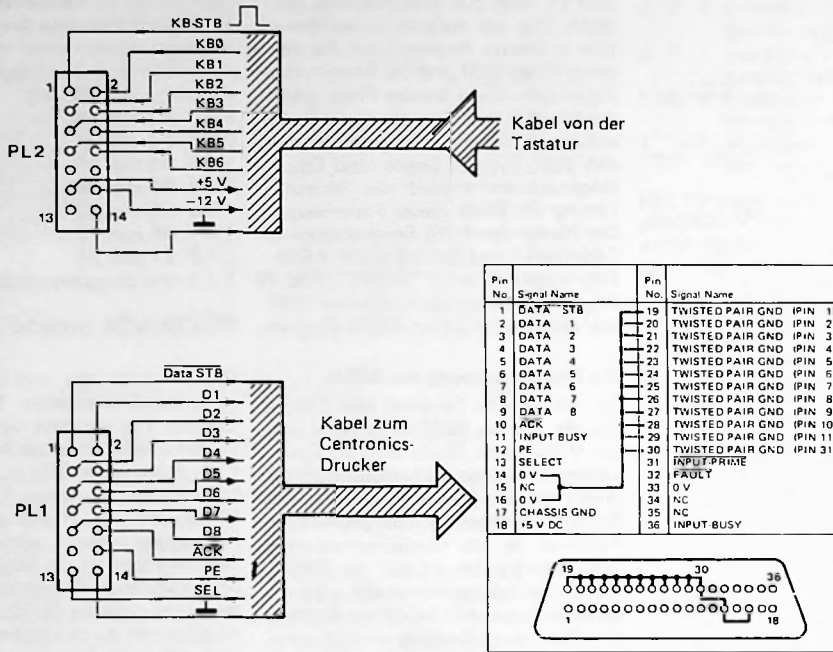
Bild 17 zeigt die serielle Schnittstelle des Universellen Terminals von Elektor. Der gesamte serielle Datenaustausch erfolgt über die Anschlußleiste PL7. Aus diesem Bild ist auch ersichtlich, wie der 25-polige RS232-Stecker (male) über ein Flachbandkabel an PL7 angeschlossen wird. Ist dieser Stecker nicht an ein Modem, sondern direkt an einen Computer angeschlossen, dann sind die Steckerpole 4-5-6-8 miteinander zu verbinden! Auf Pin 20 liegt das Data-Terminal-Ready-Signal. Über diesen Pin teilt das Terminal dem angeschlossenen Computer mit, daß es momentan nicht bereit ist, Daten zu empfangen. Pin 20 des RS232/V24 Steckers geht nach +12 Volt, wenn das Terminal bereit ist, Daten zu empfangen (negative Logik).

Anschluß einer parallelen Schnittstelle

Bild 18 zeigt, wie eine parallele Schnittstelle mit einem Flachbandkabel an PL2 angeschlossen werden kann. Die Handshake-Leitung CA1 von IC2 ist mit KB-STB (Keyboard Strobe) verbunden. Im Interrupt-Flag-Register des VIA wird ein Flag gesetzt, wenn ein positiver KB-STB kommt. Die Tastaturdaten KB0...KB6 werden im Input-Latch des VIA zwischen gespeichert. Das hat den Vorteil, daß der Wert einer gedrückten Taste nach dem Loslassen der Taste im VIA erhalten bleibt. Da auf PL2 auch +5 Volt und -12 Volt herausgeführt sind, kann die Tastatur direkt vom Universellen Terminal versorgt werden.



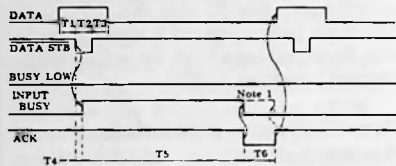
18



83903-18

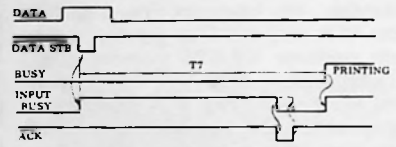
19 Timing Chart

(1) When data are accepted:



T1 ~ T3 1,4 μs min
 T4 100 ns max.
 T5 0,1 ~ 0,5 ms
 T6 6 ~ 8 μs

(2) When the data buffer is full:



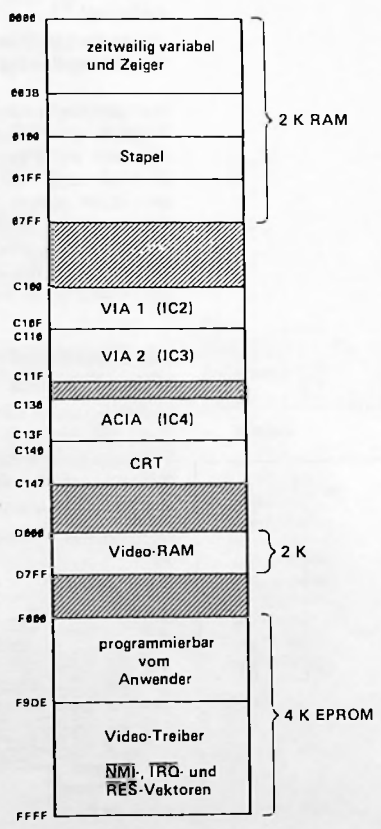
T7 = 0,2 ~ 1 ms

83903-19

Anschluß eines Druckers mit Centronics-Schnittstelle

Ein Drucker mit Centronics-Schnittstelle kann direkt an das Universelle Terminal angeschlossen werden. Alle Centronics-Signale sind auf PL1 gelegt (Bild 18). PL1 kann über ein Flachkabel mit einem Drucker mit Centronics-Schnittstelle verbunden werden. Soll der Drucker einwandfrei arbeiten, dann müssen die sogenannten Return-Leitungen "Twisted Pair GND" auf Masse gelegt werden. Bild 19 zeigt die Handshake-Signale auf dem Centronics-Steuersbus. Alle Handshake-Signale werden von der Terminal-Software überwacht und gesteuert. Bild 20 zeigt die "Memory Map" des Universellen Terminals.

20



83903-20

```

0005:
0010: F9DE          ORG    $F9DE
0015:
0020:
0025:          SOURCE LISTING OF A VT-52 COMPATIBLE TERMINAL
0030:
0035:          WRITTEN BY A. NACHTMANN
0040:
0045:
0050:          COPYRIGHT C 1983 ELEKTOR/ELEKTUUR
0055:
0060:
0065:          PROCESSOR:      6502 OR 65C02
0070:          CRT CONTROLLER: 6845 MOTOROLA, ROCKWELL, SYNERTEC
0075:
0080:
0085:          *VT-52 COMMANDS*
0090:
0095:          OTHER COMMANDS SEE COMMAND LOOKUP TABLES
0100:
0105:
0110:          CARRIAGE RETURN
0115:          LINE FEED
0120:          ESC H CURSOR HOME
0125:          ESC A CURSOR UP
0130:          ESC B CURSOR DOWN
0135:          ESC C CURSOR RIGHT
0140:          ESC D CURSOR LEFT
0145:          ESC K ERASE TO END OF LINE
0150:          ESC J ERASE TO END OF SCREEN
0155:
0160:          *POINTERS AND TEMPS*
0165:
0170: F9DE          RAMPTR *      $0000  RAMPTR
0175: F9DE          AHOLD  *      $0002  SAVE THE CHARACTER HERE
0180: F9DE          FLN    *      $0004  FIRST LINE POINTER
0185: F9DE          CLN    *      FLN    +02  CURRENT LINE POINTER
0190: F9DE          LLN    *      CLN    +02  LAST LINE POINTER
0195: F9DE          SCRPTR *      LLN    +02  SLAVE SCREEN POINTER
0200: F9DE          CURSOR *      SCRPTR +02  CURRENT CURSOR
0205: F9DE          INLINE *      CURSOR +02  IN LINE WITH WINDOW
0210: F9DE          COL    *      INLINE +01  CURRENT COLUMN
0215: F9DE          ESCFLG *      COL    +01  ESC FLAG
0220: F9DE          TEMCOL *      ESCFLG +01  SLAVE COLUMN
0225: F9DE          RAMBEG *      TEMCOL +01  THE REFRESH RAM STARTS HERE
0230: F9DE          CHAPLN *      RAMBEG +02  CHARACTER/LINE
0235: F9DE          LPSCR  *      CHAPLN +01  LINES/SCREEN
0240: F9DE          FORMAT *      LPSCR  +01  SCREEN FORMAT
0245: F9DE          INDEX  *      FORMAT +01  SOFTWARE STACK FOR CHAR. INPUT
0250: F9DE          BUFFER *      INDEX  +01
0255: F9DE          TABLE *      BUFFER +08  CRT FORMAT TABLE
0260: F9DE          JMPVEC *      TABLE +12  COMMAND ADDRESS POINTER
0265: F9DE          CENFLG *      JMPVEC +02  CENTRONICS FLAG
0270: F9DE          DUPLEX *      CENFLG +01  0=FULL; 1=HALF DUPLEX
0275: F9DE          AUTOLF *      DUPLEX +01  0=NO; 1=AUTO CRLF TO CENTRONICS
0280: F9DE          NMIVC  *      AUTOLF +01  (NMI VECTOR)

```

```

0285: F9DE          IRQVEC *      NMIVEC +02 (IRQ VECTOR)
0290:
0295:
0300:              *SPECIAL CHARACTERS*
0305:
0310: F9DE          SPACE *      $0020
0315: F9DE          ESC      *      $001B
0320:
0325:
0330:              *JUNIOR'S SERIAL I/O*
0335:
0340:
0345:
0350:
0355:              *VIA 1 ADDRESSES*
0360:
0365: F9DE          VAPBD *      $C100 PORT B DATA
0370: F9DE          VAPAD *      VAPBD +01 PORT A DATA
0375: F9DE          VAPBDD *     VAPBD +02 PORT B DATA DIRECTION
0380: F9DE          VAPADD *     VAPBD +03 PORT A DATA DIRECTION
0385: F9DE          VATACL *     VAPBD +04 T1, LATCH-LOW, COUNTER-LOW
0390: F9DE          VATACH *     VAPBD +05 T1, COUNTER-HIGH
0395: F9DE          VATALL *     VAPBD +06 T1, LATCH-LOW
0400: F9DE          VATALH *     VAPBD +07 T1, LATCH-HIGH
0405: F9DE          VATBCL *     VAPBD +08 T2, LATCH-LOW, COUNTER-LOW
0410: F9DE          VATBCH *     VAPBD +09 T2, COUNTER-HIGH
0415: F9DE          VASR *      VAPBD +0A SHIFT REGISTER
0420: F9DE          VAACR *     VAPBD +0B AUXILIARY CONTROL REGISTER
0425: F9DE          VAPCR *     VAPBD +0C PERIPHERAL CONTROL REGISTER
0430: F9DE          VAIFR *     VAPBD +0D INTERRUPT FLAG REGISTER
0435: F9DE          VAIER *     VAPBD +0E INTERRUPT ENABLE REGISTER
0440: F9DE          VAPADN *     VAPBD +0F PORT A DATA, NO HANDSHAKE
0445:
0450:              *VIA 2 ADDRESSES*
0455:
0460: F9DE          VBPBD *     $C110 PORT B DATA
0465: F9DE          VBPAD *     VBPBD +01 PORT A DATA
0470: F9DE          VBPBDD *    VBPBD +02 PORT B DATA DIRECTION
0475: F9DE          VBPADD *    VBPBD +03 PORT A DATA DIRECTION
0480: F9DE          VBTACL *    VBPBD +04 T1, LATCH-LOW, COUNTER-LOW
0485: F9DE          VBTACH *    VBPBD +05 T1, COUNTER-HIGH
0490: F9DE          VBTALL *    VBPBD +06 T1, LATCH-LOW
0495: F9DE          VBTALH *    VBPBD +07 T1, LATCH-HIGH
0500: F9DE          VBTBCL *    VBPBD +08 T2, LATCH-LOW, COUNTER-LOW
0505: F9DE          VBTBCH *    VBPBD +09 T2, COUNTER-HIGH
0510: F9DE          VBSR *     VBPBD +0A SHIFT REGISTER
0515: F9DE          VBACR *     VBPBD +0B AUXILIARY CONTROL REGISTER
0520: F9DE          VBPCR *     VBPBD +0C PERIPHERAL CONTROL REGISTER
0525: F9DE          VBIFR *     VBPBD +0D INTERRUPT FLAG REGISTER
0530: F9DE          VBIER *     VBPBD +0E INTERRUPT ENABLE REGISTER
0535: F9DE          VBPADN *     VBPBD +0F PORT A DATA, NO HANDSHAKE
0540:
0545:              *ACIA ADDRESSES*
0550:
0555: F9DE          RECREG *     $C130 RECEIVER REGISTER
0560: F9DE          TRAREG *     RECREG TRANSMITTER REGISTER

```

```

0565: F9DE ACIASR * RECREG +01 STATUS REGISTER
0570: F9DE ACICMD * RECREG +02 COMMAND REGISTER
0575: F9DE ACICTL * RECREG +03 CONTROL REGISTER
0580:
0585: *CRT ADDRESSES*
0590:
0595: F9DE AR * $C140 ADDRESS REGISTER OF THE CRT
0600: F9DE RFILE * AR +01 REGISTER FILE OF CRT
0605:
0610:
0615:
0620:
0625: *****
0630: * VT 52 COMPATIBLE TERMINAL *
0635: *****
0640:
0645:
0650: *INTERNAL 6845 REGISTER FILE*
0655:
0660: F9DE HORTOT * $0000 HORIZONTAL TOTAL-1
0665: F9DE HORDIS * $0001 HORIZONTAL DISPLAYED
0670: F9DE HSYPOS * $0002 HORIZONTAL SYNC POSITION
0675: F9DE VHSYWI * $0003 VERTICAL, HORIZONTAL SYNC WIDTH
0680: F9DE VERTOT * $0004 VERTICAL TOTAL-1
0685: F9DE VTOTAJ * $0005 VERTICAL TOTAL ADJUST
0690: F9DE VERDIS * $0006 VERTICAL DISPLAYED
0695: F9DE VSYPOS * $0007 VERTICAL SYNC POSITION
0700: F9DE MODE * $0008 INTERLACE MODE REGISTER
0705: F9DE SCANLN * $0009 CHARACTER SCAN LINES-1
0710: F9DE CURSTA * $000A CURSOR START
0715: F9DE CUREND * $000B CURSOR END
0720: F9DE DSPSTH * $000C DISPLAY START HIGH
0725: F9DE DSPSTL * $000D DISPLAY START LOW
0730: F9DE CURPOH * $000E CURSOR POSITION HIGH
0735: F9DE CURPOL * $000F CURSOR POSITION LOW
0740: F9DE LIPENH * $0010 LIGHT PEN HIGH
0745: F9DE LIPENL * $0011 LIGHT PEN LOW
0750: F9DE UPDATH * $0012 UPDATE HIGH (ROCKWELL, SYNERTEK)
0755: F9DE UPDATL * $0013 UPDATE LOW
0760: F9DE DUMMY * $001F DUMMY REGISTER
0765:
0770:
0775:
0780:
0785:
0790: ***VIDEO HANDLER***
0795:
0800: F9DE 48 VIDEO PHA SAVE THE REGISTERS
0805: F9DF 8A TXA
0810: F9E0 48 PHA
0815: F9E1 98 TYA
0820: F9E2 48 PHA
0825: F9E3 A5 02 LDA AHOLD GET THE CHARACTER
0830: F9E5 C9 1B CMPIM ESC IS IT THE ESC CHAR.?
0835: F9E7 D0 15 BNE CHECK
0840: F9E9 85 10 STA ESCFLG SET THE ESC FLAG

```

```

0845:
0850: F9EB 68          VIDEND PLA          RESTORE THE REGISTERS
0855: F9EC A8          TAY
0860: F9ED 68          PLA
0865: F9EE AA          TAX
0870: F9EF 68          PLA
0875: F9F0 EA          NOP          $2C
0880: F9F1 EA          NOP          $80 BIT SPAD CHECK FOR BREAK
                                DURING OUTPUT
0885: F9F2 EA          NOP          $FA OR $1A
0890: F9F3 EA          NOP          $10 BPL BRKTST
0895: F9F4 EA          NOP          $01
0900: F9F5 60          RTS
0905:
0910: F9F6 EA          BRKTST NOP          $2C BIT SPAD WAIT TILL KEY
                                IS RELEASED
0915: F9F7 EA          NOP          $80
0920: F9F8 EA          NOP          $FA OR $1A
0925: F9F9 EA          NOP          $10 BPL BRKTST
0930: F9FA EA          NOP          $FB
0935: F9FB EA          NOP          $6C JMP ($FA7C) GOTO BREAK HANDLER
0940: F9FC EA          NOP          $7C
0945: F9FD EA          NOP          $FA OR $1A
0950:
0955: F9FE 20 E7 FC CHECK JSR COMCOM COMPUTE THE COMMAND INDEX
0960: FA01 B0 0E          BCS VALVEC THERE WAS NO COMMAND
0965: FA03 BD 4F FD          LDAX COMADR SET THE COMMAND VECTOR
0970: FA06 85 32          STA JMPVEC
0975: FA08 E8          INX
0980: FA09 BD 4F FD          LDAX COMADR
0985: FA0C 85 33          STA JMPVEC +01
0990: FA0E 6C 32 00          JMI JMPVEC
0995:
1000: FA11 4C 7F FA VALVEC JMP VALID
1005:
1010:
1015: ***EXECUTE A VIDEO COMMAND***
1020:
1025:
1030: *CARRIAGE RETURN*
1035:
1040: FA14 A2 00          RETURN LDXIM $00
1045: FA16 86 0F          STX COL COL=0
1050: FA18 20 EE FA          JSR ADJUST ADJUST THE CURSOR
1055: FA1B 4C EB F9          JMP VIDEND
1060:
1065: *LINE FEED*
1070:
1075: FA1E 20 52 FC FEED JSR CURDN
1080: FA21 4C EB F9          JMP VIDEND
1085:
1090: *CLEAR SCREEN & HOME CURSOR*
1095:
1100: FA24 20 97 FC CLRHOM JSR HOME
1105: FA27 20 D9 FB          JSR ERTEOS
1110: FA2A 4C EB F9          JMP VIDEND
1115:
1120: *CURSOR LEFT*

```



```

1125:
1130: FA2D 20 8D FC LEFT JSR CURLFT
1135: FA30 4C EB F9 JMP VIDEND
1140:
1145: *HOME CURSOR*
1150:
1155: FA33 20 97 FC HOCU JSR HOME
1160: FA36 4C EB F9 JMP VIDEND
1165:
1170:
1175:
1180: *CURSOR UP*
1185:
1190: FA39 20 71 FC UP JSR CURUP
1195: FA3C 4C EB F9 JMP VIDEND
1200:
1205: *CURSOR DOWN*
1210:
1215: FA3F 20 52 FC DOWN JSR CURDN
1220: FA42 4C EB F9 JMP VIDEND
1225:
1230: *CURSOR RIGHT*
1235:
1240: FA45 20 92 FC RIGHT JSR CURRGT
1245: FA48 4C EB F9 JMP VIDEND
1250:
1255: *ERASE TO END OF LINE*
1260:
1265: FA4B 20 AE FB ERLNX JSR ERTEOL
1270: FA4E 4C EB F9 JMP VIDEND
1275:
1280: *ERASE TO END OF SCREEN*
1285:
1290: FA51 20 D9 FB ERSCRX JSR ERTEOS
1295: FA54 4C EB F9 JMP VIDEND
1300:
1305:
1310:
1315:
1320: *DELETE THE CURRENT LINE*
1325:
1330: FA57 A2 00 DELLIN LDXIM $00
1335: FA59 86 0F STX COL
1340: FA5B 20 35 FB JSR ADJCUR MOVE CURSOR TO COL 0
1345: FA5E 20 AE FB JSR ERTEOL
1350: FA61 4C EB F9 JMP VIDEND
1355:
1360: *TOGGLE THE CENTRONICS FLAG*
1365:
1370: FA64 A5 34 TOGGLE LDA CENFLG
1375: FA66 49 01 EORIM $01
1380: FA68 85 34 STA CENFLG
1385: FA6A 4C EB F9 JMP VIDEND
1390:
1395: *TOGGLE THE AUTO CRLF FLAG*
1400:

```

```

1405: FA6D A5 36      TOGLF  LDA   AUTOLF
1410: FA6F 49 01      EORIM $01
1415: FA71 85 36      STA   AUTOLF
1420: FA73 4C EB F9      JMP   VIDEND
1425:
1430:                  *TOGGLE THE DUPLEX FLAG*
1435:
1440: FA76 A5 35      TOGDUP LDA   DUPLEX
1445: FA78 49 01      EORIM $01
1450: FA7A 85 35      STA   DUPLEX
1455: FA7C 4C EB F9      JMP   VIDEND
1460:
1465:
1470:
1475:                  *FILTER < SPACE*
1480:
1485: FA7F A5 02      VALID  LDA   AHOLD
1490: FA81 C9 20      CMPIM '
1495: FA83 B0 03      BCS   TOSCR
1500: FA85 4C EB F9      JMP   VIDEND
1505:
1510:                  *TRANSFER TO SCREEN*
1515:
1520: FA88 20 29 FC      TOSCR  JSR   TVPUT
1525: FA8B 4C EB F9      JMP   VIDEND
1530:
1535:
1540:
1545:                  ***SUBROUTINES***
1550:
1555:
1560:                  *FIRST LINE UP*
1565:
1570: FA8E 38          FLNUP  SEC
1575: FA8F A5 04          LDA   FLN
1580: FA91 E5 14          SBC   CHAPLN  FLN=FLN-CHAPLN
1585: FA93 85 04          STA   FLN
1590: FA95 A5 05          LDA   FLN      +01
1595: FA97 E9 00          SBCIM $00
1600: FA99 29 07          ANDIM $07      MAX IS $7FF
1605: FA9B 85 05          STA   FLN      +01
1610: FA9D 60          RTS
1615:
1620:                  *CURRENT LINE UP*
1625:
1630: FA9E 38          CLNUP  SEC
1635: FA9F A5 06          LDA   CLN
1640: FAA1 E5 14          SBC   CHAPLN
1645: FAA3 85 06          STA   CLN
1650: FAA5 A5 07          LDA   CLN      +01
1655: FAA7 E9 00          SBCIM $00
1660: FAA9 29 07          ANDIM $07      MAX IS $7FF
1665: FAAB 85 07          STA   CLN      +01
1670: FAAD 60          RTS
1675:
1680:                  *LAST LINE UP*

```

```

1685:
1690: FAAE 38          LLNUP  SEC
1695: FAAF A5 08      LDA    LLN
1700: FAB1 E5 14      SBC   CHAPLN
1705: FAB3 85 08      STA   LLN
1710: FAB5 A5 09      LDA   LLN    +01
1715: FAB7 E9 00      SBCIM $00
1720: FAB9 29 07      ANDIM $07    MAX IS $7FF
1725: FAB8 85 09      STA   LLN    +01
1730: FABD 60          RTS
1735:
1740:
1745:
1750: FABE 18          FLNDN  CLC
1755: FABF A5 04      LDA   FLN
1760: FAC1 65 14      ADC   CHAPLN
1765: FAC3 85 04      STA   FLN    FLN=FLN+CHAPLN
1770: FAC5 A5 05      LDA   FLN    +01
1775: FAC7 69 00      ADCIM $00
1780: FAC9 29 07      ANDIM $07    MAX IS $7FF
1785: FACB 85 05      STA   FLN    +01
1790: FACD 60          RTS
1795:
1800:
1805:
1810: FACE 18          CLNDN  CLC
1815: FACF A5 06      LDA   CLN
1820: FAD1 65 14      ADC   CHAPLN
1825: FAD3 85 06      STA   CLN    CLN=CLN+CHAPLN
1830: FAD5 A5 07      LDA   CLN    +01
1835: FAD7 69 00      ADCIM $00
1840: FAD9 29 07      ANDIM $07    MAX IS $7FF
1845: FADB 85 07      STA   CLN    +01
1850: FADD 60          RTS
1855:
1860:
1865:
1870: FADE 18          LLNDN  CLC
1875: FADF A5 08      LDA   LLN
1880: FAE1 65 14      ADC   CHAPLN
1885: FAE3 85 08      STA   LLN    LLN=LLN+CHAPLN
1890: FAE5 A5 09      LDA   LLN    +01
1895: FAE7 69 00      ADCIM $00
1900: FAE9 29 07      ANDIM $07    MAX IS $7FF
1905: FAEB 85 09      STA   LLN    +01
1910: FAED 60          RTS
1915:
1920:
1925:
1930: FAEE A4 0F      ADJUST LDY  COL    IS COLUMN NEGATIVE?
1935: FAF0 10 1F      BPL   ADSA    BRANCH ON NO
1940: FAF2 C6 0E      DEC   INLINE  GO BACK ONE LINE
1945: FAF4 10 10      BPL   ADJU    BRANCH IF STILL ON SCREEN
1950: FAF6 20 8E FA   JSR   FLNUP   FIRST LINE UP
1955: FAF9 20 79 FB   JSR   FLNCRT  ADJUST DISPLAY START
1960: FAFC 20 07 FC   JSR   ERAFLN  ERASE THE FIRST LINE

```

1965:	FAFF	20	AE	FA		JSR	LLNUP	LAST LINE UP
1970:	FB02	A0	00			LDYIM	\$00	RESET INLINE
1975:	FB04	84	0E			STY	INLINE	
1980:								
1985:	FB06	20	9E	FA	ADJU	JSR	CLNUP	CURRENT LINE UP
1990:	FB09	A4	14			LDY	CHAPLN	
1995:	FB0B	88				DEY		COL=CHAPLN-1
2000:	FB0C	84	0F			STY	COL	
2005:	FB0E	4C	35	FB		JMP	ADJCUR	
2010:								
2015:								
2020:	FB11	C4	14		ADSA	CPY	CHAPLN	IS COL>=CHAPLN?
2025:	FB13	90	20			BCC	ADJCUR	BRANCH ON NO
2030:	FB15	A0	00			LDYIM	\$00	
2035:	FB17	84	0F			STY	COL	COL=0
2040:	FB19	E6	0E			INC	INLINE	MOVE DOWN 1 LINE
2045:	FB1B	A4	0E			LDY	INLINE	
2050:	FB1D	C4	15			CPY	LPSCR	STILL ON SCREEN?
2055:	FB1F	90	11			BCC	ADJV	IS INLINE>=LPSCR?
2060:	FB21	20	BE	FA		JSR	FLNDN	FIRST LINE DOWN
2065:	FB24	20	79	FB		JSR	FLNCRT	ADJUST DISPLAY START
2070:	FB27	20	DE	FA		JSR	LLNDN	LAST LINE DOWN
2075:	FB2A	A4	15			LDY	LPSCR	
2080:	FB2C	88				DEY		
2085:	FB2D	84	0E			STY	INLINE	INLINE=LPSCR-1
2090:	FB2F	20	18	FC		JSR	ERALLN	ERASE LAST LINE
2095:								
2100:	FB32	20	CE	FA	ADJV	JSR	CLNDN	CURRENT LINE DOWN
2105:								
2110:	FB35	A2	00		ADJCUR	LDXIM	\$00	
2115:	FB37	86	0C			STX	CURSOR	RESET CURSOR
2120:	FB39	86	0D			STX	CURSOR	+01
2125:	FB3B	A6	0E			LDX	INLINE	IS INLINE=0?
2130:	FB3D	F0	0E			BEQ	ACURX	BRANCH ON YES
2135:								
2140:	FB3F	18			ACURA	CLC		
2145:	FB40	A5	14			LDA	CHAPLN	
2150:	FB42	65	0C			ADC	CURSOR	
2155:	FB44	85	0C			STA	CURSOR	
2160:	FB46	90	02			BCC	ACURB	
2165:	FB48	E6	0D			INC	CURSOR	+01 CURSOR=INLINE*CHAPLN
2170:								
2175:	FB4A	CA			ACURB	DEX		
2180:	FB4B	D0	F2			BNE	ACURA	
2185:								
2190:	FB4D	18			ACURX	CLC		
2195:	FB4E	A5	04			LDA	FLN	
2200:	FB50	65	0C			ADC	CURSOR	
2205:	FB52	85	0C			STA	CURSOR	CURSOR=CURSOR+FLN
2210:	FB54	A5	05			LDA	FLN	+01
2215:	FB56	65	0D			ADC	CURSOR	+01
2220:	FB58	85	0D			STA	CURSOR	+01
2225:								
2230:	FB5A	18				CLC		
2235:	FB5B	A5	0F			LDA	COL	
2240:	FB5D	65	0C			ADC	CURSOR	CURSOR=CURSOR+COL

```

2245: FB5F 85 0C          STA  CURSOR
2250: FB61 90 02          BCC  ACURC
2255: FB63 E6 0D          INC  CURSOR +01
2260:
2265: FB65 A2 0E          ACURC LDXIM CURPOH
2270: FB67 A5 0D          LDA  CURSOR +01
2275: FB69 8E 40 C1       STX  AR
2280: FB6C 8D 41 C1       STA  RFILE  CURSOR---->CRT CONTROLLER
2285: FB6F E8              INX
2290: FB70 A5 0C          LDA  CURSOR
2295: FB72 8E 40 C1       STX  AR
2300: FB75 8D 41 C1       STA  RFILE
2305: FB78 60              RTS
2310:
2315:                      *FIRST LINE TO CRT*
2320:
2325: FB79 A2 0C          FLNCRT LDXIM DSPSTH
2330: FB7B A5 05          LDA  FLN      +01
2335: FB7D 8E 40 C1       STX  AR
2340: FB80 8D 41 C1       STA  RFILE
2345: FB83 E8              INX
2350: FB84 A5 04          LDA  FLN
2355: FB86 8E 40 C1       STX  AR
2360: FB89 8D 41 C1       STA  RFILE
2365: FB8C 60              RTS
2370:
2375:
2380:                      *COMPUTE THE CURRENT RAM POINTER*
2385:
2390: FB8D 18              CRAMPT CLC
2395: FB8E A5 0A          LDA  SCRPTTR
2400: FB90 65 12          ADC  RAMBEG  RAMPTR=SCRPTTR+RAMBEG
2405: FB92 85 00          STA  RAMPTR
2410: FB94 A5 0B          LDA  SCRPTTR +01
2415: FB96 65 13          ADC  RAMBEG +01
2420: FB98 29 D7          ANDIM $D7  MAX IS $D7FF
2425: FB9A 85 01          STA  RAMPTR +01
2430: FB9C 18              CLC
2435: FB9D A5 00          LDA  RAMPTR  RAMPTR=RAMBEG+SCRPTTR+TEMCOL
2440: FB9F 65 11          ADC  TEMCOL
2445: FBA1 85 00          STA  RAMPTR
2450: FBA3 90 08          BCC  CRAMP
2455: FBA5 E6 01          INC  RAMPTR +01
2460: FBA7 A5 01          LDA  RAMPTR +01
2465: FBA9 29 D7          ANDIM $D7  MAX IS $D7FF
2470: FBAB 85 01          STA  RAMPTR +01
2475:
2480: FBAD 60              CRAMP RTS
2485:
2490:                      *ERASE TO END OF LINEX
2495:
2500: FBAE A6 06          ERTEOL LDX  CLN
2505: FBB0 A4 07          LDY  CLN      +01
2510: FBB2 86 0A          STX  SCRPTTR SCRPTTR=CLN
2515: FBB4 84 0B          STY  SCRPTTR +01
2520: FBB6 A4 0F          LDY  COL      GET THE CURR. COLUMN

```

```

2525: FBB8 84 11          STY  TEMCOL  TEMCOL=COL
2530: FBBA A2 20          LDXIM '      SPACE TO X
2535: FBBC A0 00          LDYIM $00    INDEX=0
2540:
2545: FBBE 20 8D FB      EREOL JSR  CRAMPT RAMPTR=RAMBEG+SCRPTR+TEMCOL
2550:
2555: FBC1 8A              EROLX TXA              SPACE--->RAM
2560: FBC2 91 00          STAIY RAMPTR
2565: FBC4 E6 11          INC  TEMCOL  TEMCOL=TEMCOL+1
2570: FBC6 E6 00          INC  RAMPTR
2575: FBC8 D0 08          BNE  EROXX
2580: FBCA E6 01          INC  RAMPTR +01 RAMPTR=RAMPTR+1
2585: FBCC A5 01          LDA  RAMPTR +01
2590: FBCE 29 D7          ANDIM $D7    MAX IS $D7FF
2595: FBD0 85 01          STA  RAMPTR +01
2600:
2605: FBD2 A5 11          EROXX LDA  TEMCOL IS TEMCOL>=CHAPLN?
2610: FBD4 C5 14          CMP  CHAPLN
2615: FBD6 90 E9          BCC  EROLX  BRANCH ON NO
2620: FBD8 60              RTS
2625:
2630:                      *ERASE TO END OF SCREEN*
2635:
2640: FBD9 A6 0E          ERTEOS LDX  INLINE
2645: FBDB E8              INX      LAST LINE?
2650: FBDC E4 15          CPX  LPSCR
2655: FBDE F0 CE          BEQ  ERTEOL IF YES DO ONLY "EOL"
2660: FBE0 20 AE FB      JSR  ERTEOL DO "EOL" FOR THIS LINE
2665:
2670: FBE3 18              EREOS  CLC
2675: FBE4 A5 0A          LDA  SCRPTR
2680: FBE6 65 14          ADC  CHAPLN SCRPTR=SCRPTR+CHAPLN
2685: FBE8 85 0A          STA  SCRPTR OR SCREEN POINTER DOWN
2690: FBEA A5 0B          LDA  SCRPTR +01
2695: FBEC 69 00          ADCIM $00
2700: FBEE 29 07          ANDIM $07    MAX IS $7FF
2705: FBF0 85 0B          STA  SCRPTR +01
2710: FBF2 C5 09          CMP  LLN    +01 IS SCRPTR=LLN?
2715: FBF4 D0 06          BNE  EEOL
2720: FBF6 A5 0A          LDA  SCRPTR
2725: FBF8 C5 08          CMP  LLN
2730: FBFA F0 07          BEQ  EEOLB  ERASE LAST LINE AND STOP
2735:
2740: FBFC 84 11          EEOL  STY  TEMCOL  TEMCOL=0
2745: FBFE 20 BE FB      JSR  EREOL
2750: FC01 B0 E0          BCS  EREOS
2755:
2760:
2765: FC03 84 11          EEOLB STY  TEMCOL  TEMCOL=0
2770: FC05 F0 B7          BEQ  EREOL
2775:
2780:                      *ERASE THE FIRST LINE*
2785:
2790: FC07 A0 00          ERAFLN LDYIM $00
2795: FC09 84 11          STY  TEMCOL
2800: FC0B A2 20          LDXIM '

```

```

2805: FC0D A5 04          LDA   FLN
2810: FC0F 85 0A          STA   SCRPTR  SCRPTR=FLN
2815: FC11 A5 05          LDA   FLN      +01
2820: FC13 85 0B          STA   SCRPTR  +01
2825: FC15 4C BE FB      JMP   EREOL   NOW ERASE THE FIRST LINE
2830:
2835:                      *ERASE THE LAST LINE*
2840:
2845: FC18 A0 00          ERALLN LDYIM  $00
2850: FC1A 84 11          STY   TEMCOL  TEMCOL=0
2855: FC1C A2 20          LDXIM  '
2860: FC1E A5 08          LDA   LLN
2865: FC20 85 0A          STA   SCRPTR  SCRPTR=LLN
2870: FC22 A5 09          LDA   LLN      +01
2875: FC24 85 0B          STA   SCRPTR  +01
2880: FC26 4C BE FB      JMP   EREOL   NOW ERASE THE LAST LINE
2885:
2890:                      *PUT A CHARACTER ON THE SCREEN*
2895:
2900: FC29 18          TPUT  CLC
2905: FC2A A5 06          LDA   CLN
2910: FC2C 65 12          ADC   RAMBEG
2915: FC2E 85 00          STA   RAMPTR  RAMPTR=CLN+RAMBEG
2920: FC30 A5 07          LDA   CLN      +01
2925: FC32 65 13          ADC   RAMBEG  +01
2930: FC34 85 01          STA   RAMPTR  +01
2935: FC36 A0 00          LDYIM  $00
2940: FC38 18          CLC
2945: FC39 A5 0F          LDA   COL      RAMPTR=CLN+RAMBEG+COL
2950: FC3B 65 00          ADC   RAMPTR
2955: FC3D 85 00          STA   RAMPTR
2960: FC3F 90 08          BCC   TPX
2965: FC41 E6 01          INC   RAMPTR  +01
2970: FC43 A5 01          LDA   RAMPTR  +01
2975: FC45 29 D7          ANDIM $D7     MAX IS $D7FF
2980: FC47 85 01          STA   RAMPTR  +01
2985:
2990: FC49 A5 02          TPX   LDA   AHOLD
2995: FC4B 91 00          STAIY RAMPTR
3000: FC4D E6 0F          INC   COL     COL=COL+01
3005: FC4F 4C EE FA      JMP   ADJUST
3010:
3015:                      *CURSOR DOWN*
3020:
3025: FC52 E6 0E          CURDN INC   INLINE  INLINE=INLINE+1
3030: FC54 20 CE FA      JSR   CLNDN  CURRENT LINE DOWN
3035: FC57 A4 0E          LDY   INLINE
3040: FC59 C4 15          CPY   LPSCR  IS INLINE>=LPSCR?
3045: FC5B 90 11          BCC   CURDNX  BRANCH ON NO
3050: FC5D 20 BE FA      JSR   FLNDN  FIRST LINE DOWN
3055: FC60 20 79 FB      JSR   FLNCR  FIRST LINE--->CRT
3060: FC63 20 DE FA      JSR   LLNDN  LAST LINE DOWN
3065: FC66 20 18 FC      JSR   ERALLN  ERASE LAST LINE
3070: FC69 A4 15          LDY   LPSCR
3075: FC6B 88          DEY
3080: FC6C 84 0E          STY   INLINE  INLINE=LPSCR-1

```

```

3085:
3090: FC6E 4C 35 FB CURDNX JMP ADJCUR ADJUST THE CURSOR AND RETURN
3095:
3100: *CURSOR UP*
3105:
3110: FC71 C6 0E CURUP DEC INLINE INLINE=INLINE-1
3115: FC73 20 9E FA JSR CLNUP CURRENT LINE UP
3120: FC76 A4 0E LDY INLINE IS INLINE=NEGATIVE?
3125: FC78 10 10 BPL CURUPX BRANCH ON NO
3130: FC7A 20 8E FA JSR FLNUP FIRST LINE UP
3135: FC7D 20 79 FB JSR FLNCRT FLN--->CRT
3140: FC80 20 AE FA JSR LLNUP LAST LINE UP
3145: FC83 20 07 FC JSR ERAFLN ERASE THE FIRST LINE
3150: FC86 A0 00 LDYIM $00
3155: FC88 84 0E STY INLINE INLINE=0
3160:
3165: FC8A 4C 35 FB CURUPX JMP ADJCUR ADJUST THE CURSOR AND RETURN
3170:
3175: *CURSOR LEFT*
3180:
3185: FC8D C6 0F CURLFT DEC COL COL=COL-1
3190: FC8F 4C EE FA JMP ADJUST ADJUST THE CURSOR AND RETURN
3195:
3200: *CURSOR RIGHT*
3205:
3210: FC92 E6 0F CURRGT INC COL COL=COL+1
3215: FC94 4C EE FA JMP ADJUST ADJUST THE CURSOR AND RETURN
3220:
3225: *HOME CURSOR*
3230:
3235: FC97 A2 00 HOME LDXIM $00
3240: FC99 86 0E STX INLINE INLINE=0
3245: FC9B 86 0F STX COL COL=0
3250: FC9D A6 04 LDX FLN
3255: FC9F 86 06 STX CLN CLN=FLN
3260: FCA1 A6 05 LDX FLN +01
3265: FCA3 86 07 STX CLN +01
3270: FCA5 4C EE FA JMP ADJUST ADJUST THE CURSOR AND RETURN
3275:
3280: *MASTER RESET OF THE MEMORY MAPPED VDU*
3285:
3290: FCA8 A9 00 RESET LDAIM $00
3295: FCAA 85 04 STA FLN FLN=$0000
3300: FCAC 85 05 STA FLN +01
3305: FCAE 85 06 STA CLN CLN=$0000
3310: FCB0 85 07 STA CLN +01
3315: FCB2 85 08 STA LLN RESET LAST LINE
3320: FCB4 85 09 STA LLN +01
3325: FCB6 85 0F STA COL COL=0
3330: FCB8 85 0E STA INLINE INLINE=0
3335: FCBA A6 15 LDX LPSCR
3340: FCBC CA DEX X=LPSCR-1
3345:
3350: FCBD 18 RSA CLC
3355: FCBE A5 14 LDA CHAPLN
3360: FCC0 65 08 ADC LLN

```



```

3365: FCC2 85 08          STA  LLN      LLN=(LPSCR-1)*XCHAPLN
3370: FCC4 90 02          BCC  RSB
3375: FCC6 E6 09          INC  LLN      +01
3380:
3385: FCC8 CA              RSB  DEX
3390: FCC9 D0 F2          BNE  RSA
3395: FCCB 20 D7 FC      JSR  CRTINT  SET THE CRT TIMING REGISTERS
3400: FCCE 20 97 FC      JSR  HOME    HOME CURSOR
3405: FCD1 20 D9 FB      JSR  ERTEOS  CLEAR THE SCREEN
3410: FCD4 4C 79 FB      JMP  FLNCRT  FLN--->CRT AND RETURN
3415:
3420:                      *INITIALIZE THE CRT CONTROLLER*
3425:
3430: FCD7 A2 00          CRTINT LDXIM $00
3435:
3440: FCD9 8E 40 C1      CIA  STX  AR      SET THE FILE INDEX
3445: FCDC 85 20          LDAX TABLE
3450: FCDE 8D 41 C1      STA  RFILE  VALUE--->FILE
3455: FCE1 E8              INX
3460: FCE2 E0 10          CPXIM $10    SET ONLY THE TIMING REGISTERS
3465: FCE4 D0 F3          BNE  CIA
3470: FCE6 60              RTS
3475:
3480:
3485:
3490:                      *COMPUTE THE COMMAND ADDRESS INDEX*
3495:
3500: FCE7 A2 00          COMCOM LDXIM $00
3505: FCE9 A0 01          LDYIM $01
3510:
3515: FCEB BD 09 FD      COMCOA LDAX  COMTAB
3520: FCEE C5 10          CMP  ESCFLG  FIND THE COMMAND
3525: FCF0 D0 0A          BNE  COMCOB
3530: FCF2 B9 09 FD      LDAY COMTAB
3535: FCF5 C5 02          CMP  AHOLD
3540: FCF7 D0 03          BNE  COMCOB
3545: FCF9 18              CLC
3550: FCFA 90 08          BCC  COMCOC  C=0 --> X=ADDRESS INDEX
3555:
3560: FCFC E8              COMCOB INX
3565: FCFD E8              INX
3570: FCFE C8              INY
3575: FCFF C8              INY
3580: FD00 E0 36          CPXIM $36
3585: FD02 90 E7          BCC  COMCOA  C=1 --> NO COMMAND
3590:
3595: FD04 A9 00          COMCOC LDAIM $00
3600: FD06 85 10          STA  ESCFLG  RESET THE ESCAPE FLAG
3605: FD08 60              RTS
3610:
3615:
3620:                      *COMMAND TABLE*
3625:
3630: FD09 00          COMTAB = $00
3635: FD0A 00          = $0D  <CR>
3640: FD0B 00          = $00

```

3645:	FD0C 0A	=	\$0A	<LF>
3650:	FD0D 00	=	\$00	
3655:	FD0E 08	=	\$08	<BS>,<CTL-H> BACK SPACE
3660:	FD0F 18	=	\$1B	
3665:	FD10 48	=	'H	<ESC H> CURSOR HOME
3670:	FD11 1B	=	\$1B	
3675:	FD12 41	=	'A	<ESC A> CURSOR UP
3680:	FD13 1B	=	\$1B	
3685:	FD14 42	=	'B	<ESC B> CURSOR DOWN
3690:	FD15 1B	=	\$1B	
3695:	FD16 43	=	'C	<ESC C> CURSOR RIGHT
3700:	FD17 1B	=	\$1B	
3705:	FD18 44	=	'D	<ESC D> CURSOR LEFT
3710:	FD19 1B	=	\$1B	
3715:	FD1A 4B	=	'K	<ESC K> ERASE TO EOL
3720:	FD1B 1B	=	\$1B	
3725:	FD1C 4A	=	'J	<ESC J> ERASE TO EOS
3730:				
3735:	FD1D 00	COTABA =	\$00	
3740:	FD1E 0B	=	\$0B	<CTL-K> CURSOR UP
3745:	FD1F 00	=	\$00	
3750:	FD20 0C	=	\$0C	<CTL-L> CURSOR RIGHT
3755:	FD21 00	=	\$00	
3760:	FD22 11	=	\$11	<CTL-Q> ERASE TO EOS
3765:	FD23 00	=	\$00	
3770:	FD24 18	=	\$18	<CTL-X> ERASE TO EOL
3775:	FD25 00	=	\$00	
3780:	FD26 1A	=	\$1A	<CTL-Z> CLEAR SCREEN & HOME
3785:	FD27 00	=	\$00	
3790:	FD28 1E	=	\$1E	<CTL-^> CURSOR HOME
3795:	FD29 00	=	\$00	
3800:	FD2A 0A	=	\$0A	<CTL-J> CURSOR DOWN
3805:				
3810:	FD2B 00	COTABB =	\$00	
3815:	FD2C 10	=	\$10	<CTL-P> SELECT/DESELECT CENTRONICS
3820:	FD2D 1B	=	\$1B	
3825:	FD2E 52	=	'R	<ESC><R> DELETE LINE
3830:	FD2F 1B	=	\$1B	
3835:	FD30 2A	=	'X	<ESC><X> CLEAR SCREEN & HOME
3840:	FD31 1B	=	\$1B	
3845:	FD32 3A	=	':	<ESC><:> CLEAR SCREEN & HOME
3850:	FD33 1B	=	\$1B	
3855:	FD34 54	=	'T	<ESC><T> ERASE TO EOL
3860:	FD35 1B	=	\$1B	
3865:	FD36 74	=	't	<ESC><t> ERASE TO EOL
3870:	FD37 1B	=	\$1B	
3875:	FD38 59	=	'Y	<ESC><Y> ERASE TO EOS
3880:	FD39 1B	=	\$1B	
3885:	FD3A 79	=	'y	<ESC><y> ERASE TO EOS
3890:				
3895:	FD3B 00	COTABX =	\$00	
3900:	FD3C 06	=	\$06	<CTL-F> SELECT/DESELECT AUTOLF
3905:	FD3D 00	=	\$00	
3910:	FD3E 02	=	\$02	<CTL-B> SEL/DESEL HALF DUPLEX
3915:				
3920:	FD3F FF	=	\$FF	

```

3925: FD40 FF = $FF
3930: FD41 FF = $FF
3935: FD42 FF = $FF
3940: FD43 FF = $FF
3945: FD44 FF = $FF
3950: FD45 FF = $FF
3955: FD46 FF = $FF
3960: FD47 FF = $FF
3965: FD48 FF = $FF
3970: FD49 FF = $FF
3975: FD4A FF = $FF
3980: FD4B FF = $FF
3985: FD4C FF = $FF
3990: FD4D FF = $FF
3995: FD4E FF = $FF

```

4000:

4005:

COMMAND ADDRESS TABLE

4010:

```

4015: FD4F 14 COMADR = $14 CR
4020: FD50 FA = $FA
4025: FD51 1E = $1E LF
4030: FD52 FA = $FA
4035: FD53 2D = $2D CURSOR LEFT
4040: FD54 FA = $FA
4045: FD55 33 = $33 CURSOR HOME
4050: FD56 FA = $FA
4055: FD57 39 = $39 CURSOR UP
4060: FD58 FA = $FA
4065: FD59 3F = $3F CURSOR DOWN
4070: FD5A FA = $FA
4075: FD5B 45 = $45 CURSOR RIGHT
4080: FD5C FA = $FA
4085: FD5D 2D = $2D CURSOR LEFT
4090: FD5E FA = $FA
4095: FD5F 4B = $4B ERASE TO EOL
4100: FD60 FA = $FA
4105: FD61 51 = $51 ERASE TO EOS
4110: FD62 FA = $FA

```

4115:

4120:

4125:

NORMAL VIDEO COMMANDS

4130:

```

4135: FD63 39 = $39 CURSOR UP
4140: FD64 FA = $FA
4145: FD65 45 = $45 CURSOR RIGHT
4150: FD66 FA = $FA
4155: FD67 51 = $51 ERASE TO EOS
4160: FD68 FA = $FA
4165: FD69 4B = $4B ERASE TO EOL
4170: FD6A FA = $FA
4175: FD6B 24 = $24 CLEAR & HOME
4180: FD6C FA = $FA
4185: FD6D 33 = $33 CURSOR HOME
4190: FD6E FA = $FA
4195: FD6F 3F = $3F CURSOR DOWN
4200: FD70 FA = $FA

```

4205:	FD71	64	=	\$64	SELECT/DESELECT CENTRONICS
4210:	FD72	FA	=	\$FA	
4215:	FD73	57	=	\$57	DELETE LINE
4220:	FD74	FA	=	\$FA	
4225:	FD75	24	=	\$24	CLEAR & HOME
4230:	FD76	FA	=	\$FA	
4235:	FD77	24	=	\$24	"
4240:	FD78	FA	=	\$FA	
4245:	FD79	4B	=	\$4B	ERASE TO EOL
4250:	FD7A	FA	=	\$FA	
4255:	FD7B	4B	=	\$4B	"
4260:	FD7C	FA	=	\$FA	
4265:	FD7D	51	=	\$51	ERASE TO EOS
4270:	FD7E	FA	=	\$FA	
4275:	FD7F	51	=	\$51	"
4280:	FD80	FA	=	\$FA	
4285:	FD81	6D	=	\$6D	SELECT/DESELECT AUTOLF
4290:	FD82	FA	=	\$FA	
4295:	FD83	76	=	\$76	SEL/DESEL HALF DUPLEX
4300:	FD84	FA	=	\$FA	
4305:					
4310:	FD85	FF	=	\$FF	
4315:	FD86	FF	=	\$FF	
4320:	FD87	FF	=	\$FF	
4325:	FD88	FF	=	\$FF	
4330:	FD89	FF	=	\$FF	
4335:	FD8A	FF	=	\$FF	
4340:	FD8B	FF	=	\$FF	
4345:	FD8C	FF	=	\$FF	
4350:	FD8D	FF	=	\$FF	
4355:	FD8E	FF	=	\$FF	
4360:	FD8F	FF	=	\$FF	
4365:	FD90	FF	=	\$FF	
4370:	FD91	FF	=	\$FF	
4375:	FD92	FF	=	\$FF	
4380:	FD93	FF	=	\$FF	
4385:	FD94	FF	=	\$FF	
4390:					
4395:					
4400:					
4405:					×CENTRONICS OUTPUT×
4410:					
4415:	FD95	AD 10 C1	CENTRO	LDA	VBPBD
4420:	FD98	29 08		ANDIM	\$08
4425:	FD9A	F0 29		BEQ	CTROB
4430:	FD9C	AD 10 C1		LDA	VBPBD
4435:	FD9F	29 10		ANDIM	\$10
4440:	FDA1	D0 22		BNE	CTROB
4445:					
4450:	FDA3	A5 02	CTROA	LDA	AHOLD
4455:	FDA5	8D 00 C1		STA	VAPBD
4460:					OUTPUT THE CHARACTER AND
4465:	FDA8	AD 10 C1	WAIT	LDA	VBPBD
4470:	FDAB	29 08		ANDIM	\$08
4475:	FDAD	F0 16		BEQ	CTROB
4480:	FDAF	AD 10 C1		LDA	VBPBD
					SEL ?

```

4485: FDB2 29 10          ANDIM $10    PE ?
4490: FDB4 D0 0F          BNE    CTROB
4495: FDB6 AD 0D C1       LDA    VAIFR  WAIT FOR ACKNOWLEDGE
4500: FDB9 29 10          ANDIM $10    SAMPLE THE CB1 FLAG
4505: FDBB F0 EB          BEQ    WAIT
4510: FDBD AD 0D C1       LDA    VAIFR  RESET CB1 & CB2 FLAGS
4515: FDC0 09 18          ORAIM $18
4520: FDC2 8D 0D C1       STA    VAIFR
4525:
4530: FDC5 60              CTROB  RTS
4535:
4540:
4545:
4550:                    *INITIALIZE CENTRONICS*
4555:
4560: FDC6 A9 FF          INICEN LDAIM $FF    PB0...PB7 = OUTPUT
4565: FDC8 8D 02 C1       STA    VAPBDD
4570: FDCB A9 A0          LDAIM $A0    CB2 = WRITE HANDSHAKE PULSE OUTPUT
4575: FDCD 8D 0C C1       STA    VAPCR  CB1 = NEG. EDGE SENSITIVE
4580: FDD0 AD 12 C1       LDA    VBPBDD
4585: FDD3 29 E7          ANDIM $E7    PB3,PB4 = INPUT
4590: FDD5 8D 12 C1       STA    VBPBDD
4595: FDD8 A9 00          LDAIM $00    RESET THE PRINTER FLAG
4600: FDDA 85 34          STA    CENFLG
4605: FDDC 60              RTS
4610:
4615:                    *MOVE THE CRT FILE FROM ROM TO RAM*
4620:
4625: FDDD A9 00          MOVCR   LDAIM $00    REFRESH RAM STARTS AT $D000
4630: FDDF A2 D0          LDXIM $D0
4635: FDE1 85 12          STA    RAMBEG
4640: FDE3 86 13          STX    RAMBEG +01
4645: FDE5 A9 00          LDAIM $00
4650: FDE7 A8              TAY
4655: FDE8 A6 16          LDX    FORMAT  GET THE CURR. FORMAT
4660: FDEA F0 06          BEQ    MCRTB
4665:
4670: FDEC 18              MCRTA  CLC
4675: FDED 69 12          ADCIM $12    COMPUTE THE INDEX
4680: FDEF CA              DEX
4685: FDF0 D0 FA          BNE    MCRTA
4690:
4695: FDF2 AA              MCRTB  TAX
4700:
4705: FDF3 BD 12 FE       MCRTC  LDAX    CRTINA
4710: FDF6 99 20 00       STAY   TABLE  MOVE TABLE
4715: FDF9 E8              INX
4720: FDFA C8              INY
4725: FDFB C0 12          CPYIM $12
4730: FDFD D0 F4          BNE    MCRTC
4735: FDFF 88              DEY
4740: FE00 B9 20 00       LDAY   TABLE  SET SCREEN PARAMETERS
4745: FE03 85 15          STA    LPSCR
4750: FE05 88              DEY
4755: FE06 B9 20 00       LDAY   TABLE
4760: FE09 85 14          STA    CHAPLN

```

```

4765: FE0B 60          RTS
4770:
4775:          *INDIRECT NMI & IRQ VECTORS*
4780:
4785: FE0C 6C 37 00  NMIJMP JMI  NMIVEC
4790: FE0F 6C 39 00  IRQJMP JMI  IRQVEC
4795:
4800:
4805:          *CRT TIMING TABLES*
4810:
4815:          - 80x24 -
4820:
4825: FE12 80          CRTINA =      $80 HORIZONTAL TOTAL-1 = 129-1 CHAR.
4830: FE13 50          =      $50 HORIZONTAL DISPLAYED = 80 CHAR.
4835: FE14 60          =      $60 HORIZ. SYNC. POSITION = 96 CHAR.
4840: FE15 08          =      $08 VERT./HORIZ. SYNC WIDTH = 16/8
4845: FE16 22          =      $22 VERTICAL TOTAL-1 = 34 CHAR. LINES
4850: FE17 00          =      $00 VERT. TOTAL ADJ. = 0x64 MICRO SEC.
4855: FE18 18          =      $18 VERTICAL DISPLAYED = 24 LINES
4860: FE19 1C          =      $1C VERT. SYNC. POSITION = 29 CHAR. LINES
4865: FE1A 00          =      $00 MODE CONTROL
4870: FE1B 08          =      $08 SCAN LINES-1 = 9-1
4875: FE1C 00          =      $00 CURSOR START
4880: FE1D 09          =      $09 CURSOR END
4885: FE1E 00          =      $00 DISPLAY START (NOT NEEDED)
4890: FE1F 00          =      $00
4895: FE20 00          =      $00 CURSOR POSITION (NOT NEEDED)
4900: FE21 00          =      $00
4905: FE22 50          =      $50 CHARACTERS/LINE
4910: FE23 18          =      $18 LINES/SCREEN
4915:
4920:
4925:          - 80x25 -
4930:
4935: FE24 80          =      $80
4940: FE25 50          =      $50
4945: FE26 60          =      $60
4950: FE27 08          =      $08
4955: FE28 22          =      $22
4960: FE29 00          =      $00
4965: FE2A 19          =      $19
4970: FE2B 1C          =      $1C
4975: FE2C 00          =      $00
4980: FE2D 08          =      $08
4985: FE2E 00          =      $00
4990: FE2F 09          =      $09
4995: FE30 00          =      $00
5000: FE31 00          =      $00
5005: FE32 00          =      $00
5010: FE33 00          =      $00
5015: FE34 50          =      $50
5020: FE35 19          =      $19
5025:
5030:          - 64x16 -
5035:
5040: FE36 64          =      $64

```

5045:	FE37	40	=	\$40
5050:	FE38	49	=	\$49
5055:	FE39	05	=	\$05
5060:	FE3A	16	=	\$16
5065:	FE3B	0E	=	\$0E
5070:	FE3C	10	=	\$10
5075:	FE3D	12	=	\$12
5080:	FE3E	00	=	\$00
5085:	FE3F	0C	=	\$0C
5090:	FE40	00	=	\$00
5095:	FE41	09	=	\$09
5100:	FE42	00	=	\$00
5105:	FE43	00	=	\$00
5110:	FE44	00	=	\$00
5115:	FE45	00	=	\$00
5120:	FE46	40	=	\$40
5125:	FE47	10	=	\$10
5130:				
5135:			- 64x24 -	
5140:				
5145:	FE48	64	=	\$64
5150:	FE49	40	=	\$40
5155:	FE4A	52	=	\$52
5160:	FE4B	05	=	\$05
5165:	FE4C	22	=	\$22
5170:	FE4D	00	=	\$00
5175:	FE4E	18	=	\$18
5180:	FE4F	1C	=	\$1C
5185:	FE50	00	=	\$00
5190:	FE51	08	=	\$08
5195:	FE52	00	=	\$00
5200:	FE53	09	=	\$09
5205:	FE54	00	=	\$00
5210:	FE55	00	=	\$00
5215:	FE56	00	=	\$00
5220:	FE57	00	=	\$00
5225:	FE58	40	=	\$40
5230:	FE59	18	=	\$18
5235:				
5240:			- 90x22 -	
5245:				
5250:	FE5A	87	=	\$87
5255:	FE5B	5A	=	\$5A
5260:	FE5C	6A	=	\$6A
5265:	FE5D	08	=	\$08
5270:	FE5E	22	=	\$22
5275:	FE5F	00	=	\$00
5280:	FE60	16	=	\$16
5285:	FE61	1C	=	\$1C
5290:	FE62	00	=	\$00
5295:	FE63	08	=	\$08
5300:	FE64	00	=	\$00
5305:	FE65	09	=	\$09
5310:	FE66	00	=	\$00
5315:	FE67	00	=	\$00
5320:	FE68	00	=	\$00

5325:	FE69	00	=	\$00
5330:	FE6A	5A	=	\$5A
5335:	FE6B	16	=	\$16
5340:				
5345:			- 48x12 -	
5350:				
5355:	FE6C	46	=	\$46
5360:	FE6D	30	=	\$30
5365:	FE6E	3A	=	\$3A
5370:	FE6F	05	=	\$05
5375:	FE70	16	=	\$16
5380:	FE71	0E	=	\$0E
5385:	FE72	0C	=	\$0C
5390:	FE73	12	=	\$12
5395:	FE74	00	=	\$00
5400:	FE75	0C	=	\$0C
5405:	FE76	00	=	\$00
5410:	FE77	09	=	\$09
5415:	FE78	00	=	\$00
5420:	FE79	00	=	\$00
5425:	FE7A	00	=	\$00
5430:	FE7B	00	=	\$00
5435:	FE7C	30	=	\$30
5440:	FE7D	0C	=	\$0C
5445:				
5450:			- 24x24 -	
5455:				
5460:	FE7E	38	=	\$38
5465:	FE7F	18	=	\$18
5470:	FE80	26	=	\$26
5475:	FE81	05	=	\$05
5480:	FE82	22	=	\$22
5485:	FE83	00	=	\$00
5490:	FE84	18	=	\$18
5495:	FE85	1C	=	\$1C
5500:	FE86	00	=	\$00
5505:	FE87	08	=	\$08
5510:	FE88	00	=	\$00
5515:	FE89	09	=	\$09
5520:	FE8A	00	=	\$00
5525:	FE8B	00	=	\$00
5530:	FE8C	00	=	\$00
5535:	FE8D	00	=	\$00
5540:	FE8E	18	=	\$18
5545:	FE8F	18	=	\$18
5550:				
5555:			- 80x24 (USA) -	
5560:				
5565:	FE90	00	=	\$00
5570:	FE91	00	=	\$00
5575:	FE92	00	=	\$00
5580:	FE93	00	=	\$00
5585:	FE94	00	=	\$00
5590:	FE95	00	=	\$00
5595:	FE96	00	=	\$00
5600:	FE97	00	=	\$00

5605:	FE98	00	=	\$00	
5610:	FE99	00	=	\$00	
5615:	FE9A	00	=	\$00	
5620:	FE9B	00	=	\$00	
5625:	FE9C	00	=	\$00	
5630:	FE9D	00	=	\$00	
5635:	FE9E	00	=	\$00	
5640:	FE9F	00	=	\$00	
5645:	FEA0	50	=	\$50	
5650:	FEA1	18	=	\$18	
5655:					
5660:					
5665:					
5670:	FEA2	D8	RESVTA	CLD	
5675:	FEA3	78		SEI	
5680:	FEA4	A2 FF		LDXIM \$FF	RESET THE STACK POINTER
5685:	FEA6	9A		TXS	
5690:	FEA7	20 63 FF		JSR INIKBD	INIT. PAR. KEYBOARD
5695:	FEAA	20 85 FF		JSR CTLCMD	READ ACIA CONTROL/COMMAND REG.
5700:	FEAD	20 AE FF		JSR GETFOT	GET THE SCREEN FORMAT
5705:	FEB0	20 DD FD		JSR MOVCRT	MOVE THE CRT TABLE & SCREEN FORMAT
5710:	FEB3	20 A8 FC		JSR RESET	INIT. CRT
5715:	FEB6	AD 10 C1		LDA VBPBD	SET READY LOW
5720:	FEB9	29 FE		ANDIM \$FE	
5725:	FEBB	8D 10 C1		STA VBPBD	
5730:	FEBE	AD 12 C1		LDA VBPBDD	
5735:	FEC1	09 01		ORAIM \$01	PB0=READY-OUTPUT
5740:	FEC3	8D 12 C1		STA VBPBDD	
5745:	FEC6	A9 00		LDAIM \$00	
5750:	FEC8	85 17		STA INDEX	NO CHARACTERS IN BUFFER
5755:	FECA	85 18		STA BUFFER	
5760:	FECC	A9 44		LDAIM RECIRQ	SET IRQ VECTOR
5765:	FECE	A2 FF		LDXIM RECIRQ	/
5770:	FED0	85 39		STA IRQVEC	
5775:	FED2	86 3A		STX IRQVEC	+01
5780:	FED4	20 C6 FD		JSR INICEN	INITIALIZE CENTRONICS I/O
5785:	FED7	A9 00		LDAIM \$00	
5790:	FED9	A2 00		LDXIM \$00	
5795:	FEDB	85 35		STA DUPLEX	SELECT FULL DUPLEX
5800:	FEDD	86 36		STX AUTOLF	OUTPUT NO AUTO LF TO CENTRONICS
5805:	FEDF	58		CLI	
5810:					
5815:	FEE0	20 76 FF	VTA	JSR GETKBD	
5820:	FEE3	D0 2C		BNE VTAT	ACTIVE KEY?
5825:	FEE5	A5 18		LDA BUFFER	
5830:	FEE7	F0 F7		BEQ VTA	ANY CHARACTERS IN BUFFER?
5835:	FEE9	A2 00		LDXIM \$00	
5840:					
5845:	FEEB	E8	VTAS	INX	
5850:	FEEC	B5 18		LDAX BUFFER	
5855:	FEED	85 02		STA AHOLD	
5860:	FEF0	20 DE F9		JSR VIDEO	
5865:	FEF3	A5 34		LDA CENFLG	
5870:	FEF5	F0 03		BEQ VTACEN	
5875:	FEF7	20 25 FF		JSR SPECEN	OUTPUT THE CHAR. TO THE PRINTER
5880:					

```

5885: FEFA E4 17      VTACEN CPX   INDEX
5890: FEFC D0 ED      BNE   UTAS
5895: FEFE 78          SEI
5900: FEFF A2 00      LDXIM $00
5905: FF01 86 18      STX   BUFFER
5910: FF03 86 17      STX   INDEX
5915: FF05 58          CLI
5920: FF06 AD 10 C1   LDA   VBPBD
5925: FF09 29 FE      ANDIM $FE
5930: FF0B 8D 10 C1   STA   VBPBD   VIA2,PB0 = OUTPUT
5935: FF0E 4C E0 FE   JMP   VTA
5940:
5945: FF11 20 D1 FF   VTAT   JSR   ACIOUT
5950: FF14 A5 35      LDA   DUPLEX
5955: FF16 F0 03      BEQ   VTATXY 0=FULL DUPLEX
5960: FF18 20 DE F9   JSR   VIDEO  ECHO TO SCREEN
5965:
5970: FF1B A5 34      VTATXY LDA   CENFLG
5975: FF1D F0 C1      BEQ   VTA   OUTPUT TO CENTRONICS?
5980: FF1F 20 25 FF   JSR   SPECEN
5985: FF22 4C E0 FE   JMP   VTA
5990:
5995:
6000:
6005:
6010:
6015:
6020:
6025: FF25 A5 36      SPECEN LDA   AUTOLF
6030: FF27 F0 17      BEQ   VTATZX ADD AN AUTOLF?
6035: FF29 A5 02      LDA   AHOLD
6040: FF2B C9 0D      CMPIM $0D   CHECK FOR CR
6045: FF2D D0 11      BNE   VTATZX
6050: FF2F 20 95 FD   JSR   CENTRO OUTPUT A CR
6055: FF32 A5 02      LDA   AHOLD
6060: FF34 48          PHA
6065: FF35 A9 0A      LDAIM $0A
6070: FF37 85 02      STA   AHOLD
6075: FF39 20 95 FD   JSR   CENTRO OUTPUT A LF
6080: FF3C 68          PLA
6085: FF3D 85 02      STA   AHOLD
6090: FF3F 60          RTS
6095:
6100:
6105: FF40 20 95 FD   VTATZX JSR   CENTRO OUTPUT THE CURR. CHAR.
6110: FF43 60          RTS
6115:
6120: FF44 48          RECIRQ PHA
6125: FF45 AD 10 C1   LDA   VBPBD
6130: FF48 09 01      ORAIM $01   READY=HIGH
6135: FF4A 8D 10 C1   STA   VBPBD
6140: FF4D 8A          TXA
6145: FF4E 48          PHA
6150: FF4F AD 31 C1   LDA   ACIASR RESET THE IRQ LINE
6155: FF52 E6 17      INC   INDEX
6160: FF54 A6 17      LDX   INDEX

```

```

6165: FF56 AD 30 C1      LDA   RECREG GET THE CHARACTER
6170: FF59 95 18        STAX  BUFFER AND STORE IT IN THE FIFO
6175: FF5B A9 01        LDAIM $01
6180: FF5D 85 18        STA   BUFFER THERE ARE CHAR. IN THE FIFO
6185: FF5F 68          PLA
6190: FF60 AA          TAX
6195: FF61 68          PLA
6200: FF62 40          RTI
6205:
6210:
6215:
6220:
6225:
6230:
6235: FF63 A9 01      INIKBD LDAIM $01      ENABLE INPUT LATCHING
6240: FF65 8D 0B C1    STA   VAACR
6245: FF68 AD 0C C1    LDA   VAPCR
6250: FF6B 09 01      ORAIM $01      CA1 = ACTIVE POSITIVE EDGE
6255: FF6D 8D 0C C1    STA   VAPCR
6260: FF70 A9 00      LDAIM $00      PA0...PA7=INPUT
6265: FF72 8D 03 C1    STA   VAPADD
6270: FF75 60          RTS
6275:
6280:
6285:
6290:
6295: FF76 AD 0D C1    GETKBD LDA   VAIFR
6300: FF79 29 02      ANDIM $02      CA1 FLAG = 1?
6305: FF7B F0 07      BEQ   GETKA
6310: FF7D AD 01 C1    LDA   VAPAD    READ KEYBOARD DATA AND
6315: FF80 29 7F      ANDIM $7F
6320: FF82 85 02      STA   AHOLD    RESET CA1 FLAG
6325:
6330: FF84 60          GETKA  RTS
6335:
6340:
6345:
6350:
6355:
6360: FF85 A9 02      CTLCMD LDAIM $02      VBPB1 = OUTPUT
6365: FF87 8D 12 C1    STA   VBPBDD
6370: FF8A A2 00      LDXIM $00      VBPB1 = LOW
6375: FF8C 8E 10 C1    STX   VBPBD
6380: FF8F AD 11 C1    LDA   VBPAD    GET ACICMD
6385: FF92 49 FF      EORIM $FF
6390: FF94 8D 32 C1    STA   ACICMD
6395: FF97 A9 04      LDAIM $04      VBPB2 = OUTPUT
6400: FF99 8D 12 C1    STA   VBPBDD
6405: FF9C 8E 10 C1    STX   VBPBD    VBPB2 = LOW
6410: FF9F AD 11 C1    LDA   VBPAD    GET ACICTL
6415: FFA2 49 FF      EORIM $FF
6420: FFA4 8D 33 C1    STA   ACICTL
6425: FFA7 8E 13 C1    STX   VBPADD   VBPAD = INPUT
6430: FFAA 8E 12 C1    STX   VBPBDD   VBPBD = INPUT
6435: FFAD 60          RTS
6440:

```

```

6445:
6450:
6455:          *GET FORMAT*
6460:
6465: FFAE AD 12 C1  GETFOT LDA   VBPBDD
6470: FFB1 29 1F          ANDIM $1F   PB7...PB5 = INPUT
6475: FFB3 8D 12 C1          STA   VBPBDD
6480: FFB6 AD 10 C1          LDA   VBPBD
6485: FFB9 49 FF          EORIM $FF
6490: FFBB A2 05          LDXIM $05
6495:
6500: FFBD 4A          GETFA  LSRA
6505: FFBE CA          DEX
6510: FFBF D0 FC          BNE   GETFA
6515: FFC1 85 16          STA   FORMAT
6520: FFC3 60          RTS
6525:
6530:
6535:
6540:          *ACIA RECEIVER*
6545:
6550: FFC4 A9 08          ACIIN  LDAIM $08   RECEIVER MASK
6555: FFC6 2D 31 C1          AND   ACIASR RECEIVER FULL?
6560: FFC9 F0 F9          BEQ   ACIIN
6565: FFCB AD 30 C1          LDA   RECREG
6570: FFCE 85 02          STA   AHOLD  READ & SAVE THE CHAR.
6575: FFD0 60          RTS
6580:
6585:
6590:          ACIA TRANSMITTER*
6595:
6600: FFD1 A9 10          ACIOUT LDAIM $10   TRANSMITTER MASK
6605: FFD3 2D 31 C1          AND   ACIASR TRANSMITTER EMPTY?
6610: FFD6 F0 F9          BEQ   ACIOUT
6615: FFD8 A5 02          LDA   AHOLD
6620: FFDA 8D 30 C1          STA   TRAREG GET AND TRANSMIT THE CHAR.
6625: FFDD 60          RTS
6630:
6635:
6640:          *CHECK TERMINAL ROUTINE*
6645:
6650: FFDE D8          CHE   CLD
6655: FFDF 78          SEI
6660: FFE0 A2 FF          LDXIM $FF
6665: FFE2 9A          TXS
6670: FFE3 20 63 FF          JSR   INIKBD
6675: FFE6 20 AE FF          JSR   GETFOT
6680: FFE9 20 DD FD          JSR   MOVCRD
6685: FFEC 20 A8 FC          JSR   RESET
6690:
6695: FFEF 20 76 FF          CHECKA JSR   GETKBD
6700: FFF2 F0 FB          BEQ   CHECKA
6705: FFF4 20 DE F9          JSR   VIDEO
6710: FFF7 4C EF FF          JMP   CHECKA
6715:
6720:
-T

```

SYMBOL TABLE 3400 386E

HORTOT	0000	RAMPTR	0000	HORDIS	0001	AHOLD	0002
HSYPOS	0002	VHSYWI	0003	FLN	0004	VERTOT	0004
VTOTAJ	0005	CLN	0006	VERDIS	0006	VSYPOS	0007
LLN	0008	MODE	0008	SCANLN	0009	CURSTA	000A
SCRPTR	000A	CUREND	000B	CURSOR	000C	DSPSTH	000C
DSPSTL	000D	CURPOH	000E	INLINE	000E	COL	000F
CURPOL	000F	ESCFLG	0010	LIPENH	0010	LIPENL	0011
TEMCOL	0011	RAMBEG	0012	UPDATH	0012	UPDATL	0013
CHAPLN	0014	LPSCR	0015	FORMAT	0016	INDEX	0017
BUFFER	0018	ESC	001B	DUMMY	001F	SPACE	0020
TABLE	0020	JMPVEC	0032	CENFLG	0034	DUPLEX	0035
AUTOLF	0036	NMIVEC	0037	IRQVEC	0039	VAPBD	C100
VAPAD	C101	VAPBDD	C102	VAPADD	C103	VATACL	C104
VATACH	C105	VATALL	C106	VATALH	C107	VATBCL	C108
VATBCH	C109	VASR	C10A	VAACR	C10B	VAPCR	C10C
VAIFR	C10D	VAIER	C10E	VAPADN	C10F	VBPBD	C110
VBPAD	C111	VBPBDD	C112	VBPADD	C113	VBTACL	C114
VBTACH	C115	VBTALL	C116	VBTALH	C117	VBTBCL	C118
VBTBCH	C119	VBSR	C11A	VBACR	C11B	VBPCR	C11C
VBIFR	C11D	VBIER	C11E	VBPADN	C11F	RECREG	C130
TRAREG	C130	ACIASR	C131	ACICMD	C132	ACICTL	C133
AR	C140	RFILE	C141	VIDEO	F9DE	VIDEND	F9EB
BRKTST	F9F6	CHECK	F9FE	VALVEC	FA11	RETURN	FA14
FEED	FA1E	CLRHOM	FA24	LEFT	FA2D	HOCU	FA33
UP	FA39	DOWN	FA3F	RIGHT	FA45	ERLNK	FA4B
ERSCRX	FA51	DELLIN	FA57	TOGGLE	FA64	TOGLF	FA6D
TOGDUP	FA76	VALID	FA7F	TOSCR	FA88	FLNUP	FA8E
CLNUP	FA9E	LLNUP	FAAE	FLNDN	FABE	CLNDN	FACE
LLNDN	FADE	ADJUST	FAEE	ADJU	FB06	ADSA	FB11
ADJV	FB32	ADJCUR	FB35	ACURA	FB3F	ACURB	FB4A
ACURX	FB4D	ACURC	FB65	FLNCRT	FB79	CRAMPT	FB8D
CRAMP	FBAD	ERTEOL	FBAE	EREOL	FBBE	EROLX	FBC1
ERQXX	FBD2	ERTEOS	FB09	EREOS	FBE3	EEOL	FBFC
EEOLB	FC03	ERAFLN	FC07	ERALLN	FC18	TVPUT	FC29
TPX	FC49	CURDN	FC52	CURDNX	FC6E	CURUP	FC71
CURUPX	FC8A	CURLFT	FC8D	CURRGT	FC92	HOME	FC97
RESET	FCA8	RSA	FCBD	RSB	FCC8	CRTINT	FCD7
CIA	FCD9	COMCOM	FCE7	COMCOA	FCEB	COMCOB	FCFC
COMCOC	FD04	COMTAB	FD09	COTABA	FD1D	COTABB	FD2B
COTABX	FD3B	COMADR	FD4F	CENTRO	FD95	CTROA	FDA3
WAIT	FDA8	CTROB	FDC5	INICEN	FDC6	MOVCRT	FDDD
MCRTA	FDEC	MCRTB	FDF2	MCRTC	FDF3	NMIJMP	FE0C
IRQJMP	FE0F	CRTINA	FE12	RESVTA	FEA2	VTA	FEE0
VTAS	FEEB	VTACEN	FEFA	VTAT	FF11	VTATXY	FF1B
SPECEN	FE25	VTATZX	FF40	RECIRQ	FF44	INIKBD	FF63
GETKBD	FF76	GETKA	FF84	CTLCMD	FF85	GETFOT	FFAE
GETFA	FFBD	ACIIN	FFC4	ACIOUT	FFD1	CHE	FFDE
CHECKA	FFEF						

SYMBOL TABLE 3400 386E

ACIASR	C131	ACICMD	C132	ACICTL	C133	ACIIN	FFC4
ACIOUT	FFD1	ACURA	FB3F	ACURB	FB4A	ACURC	FB65
ACURX	FB4D	ADJCUR	FB35	ADJU	FB06	ADJUST	FAEE
ADJV	FB32	ADSA	FB11	AHOLD	0002	AR	C140
AUTOLF	0036	BRKTST	F9F6	BUFFER	0018	CENFLG	0034
CENTRO	FD95	CHAPLN	0014	CHECK	F9FE	CHECKA	FFEF
CHE	FFDE	CIA	FCD9	CLNDN	FACE	CLNUP	FA9E
CLN	0006	CLRHOM	FA24	COL	000F	COMADR	FD4F
COMCOA	FCEB	COMCOB	FCFC	COMCOC	FD04	COMCOM	FCE7
COMTAB	FD09	CQTABA	FD1D	COTABB	FD2B	COTABX	FD3B
CRAMP	FBAD	CRAMPT	FB8D	CRTINA	FE12	CRTINT	FCD7
CTLCMD	FF85	CTROA	FDA3	CTROB	FDC5	CURDN	FC52
CURDNX	FC6E	CUREND	000B	CURLFT	FC8D	CURPOH	000E
CURPOL	000F	CURRGT	FC92	CURSOR	000C	CURSTA	000A
CURUP	FC71	CURUPX	FC8A	DELLIN	FA57	DOWN	FA3F
DSPSTH	000C	DSPSTL	000D	DUMMY	001F	DUPLEX	0035
EEOL	FBFC	EEOLB	FC03	ERAFLN	FC07	ERALLN	FC18
EEOLE	FBBE	EREOS	FBE3	ERLNX	FA4B	EROLX	FBC1
EROXX	FBD2	ERSCRX	FA51	ERTEOL	FBAE	ERTEOS	FBD9
ESCFLG	0010	ESC	001B	FEED	FA1E	FLNCRT	FB79
FLNDN	FABE	FLNUP	FA8E	FLN	0004	FORMAT	0016
GETFA	FFBD	GETFOT	FFAE	GETKA	FF84	GETKBD	FF76
HOCU	FA33	HOME	FC97	HORDIS	0001	HORTOT	0000
HSYPOS	0002	INDEX	0017	INICEN	FDC6	INIKBC	FF63
INLINE	000E	IRQJMP	FE0F	IRQVEC	0039	JMPVEC	0032
LEFT	FA2D	LIPENH	0010	LIPENL	0011	LLNDN	FADE
LLNUP	FAAE	LLN	0008	LPSCR	0015	MCRTA	FDEC
MCRTB	FDF2	MCRTC	FDF3	MODE	0008	MOVCRT	FDDD
NMIJMP	FE0C	NMIVEC	0037	RAMBEG	0012	RAMPTR	0000
RECIRQ	FF44	RECREG	C130	RESET	FCA8	RESVTA	FEA2
RETURN	FA14	RFILE	C141	RIGHT	FA45	RSA	FCBD
RSB	FCC8	SCANLN	0009	SCRPTR	000A	SPACE	0020
SPECEN	FF25	TABLE	0020	TEMCOL	0011	TOGDUP	FA76
TOGGLE	FA64	TOGLF	FA6D	TOSCR	FA88	TPX	FC49
TRAREG	C130	TVPUT	FC29	UP	FA39	UPDATH	0012
UPDATL	0013	VAACR	C10B	VAIER	C10E	VAIFR	C10D
VALID	FA7F	VALVEC	FA11	VAPAD	C101	VAPADD	C103
VAPADN	C10F	VAPBD	C100	VAPBDD	C102	VAPCR	C10C
VASR	C10A	VATACH	C105	VATACL	C104	VATALH	C107
VATALL	C106	VATBCH	C109	VATBCL	C108	VBACR	C11B
VBIER	C11E	VBIFR	C11D	VBPAD	C111	VBPADD	C113
VBPADN	C11F	VBPBD	C110	VBPBDD	C112	VBPCR	C11C
VBSR	C11A	VBTACH	C115	VBTACL	C114	VBTALH	C117
VBTALL	C116	VBTBCH	C119	VBTBCL	C118	VERDIS	0006
VERTOT	0004	VHSYWI	0003	VIDEND	F9EB	VIDEO	F9DE
VSYPOS	0007	VTACEN	FEFA	VTAS	FEEB	VTAT	FF11
VTATXY	FF1B	VTATZX	FF40	VTA	FEE0	VTOTAJ	0005
WAIT	FDA8						

