

2. Teil: Software

Terminal- Computer

Christian Persson

Was unterscheidet ein 'dummes' von einem intelligenten Video-Terminal? Nun, das eine braucht einen Berg von TTL-Bausteinen, um auch nur die Basis-Funktionen wie 'Bildschirm löschen', 'Zeilenvorschub' oder 'Backspace' zustande zu bringen. Das andere unterstützt einen komplexeren Bildaufbau durch eine Vielzahl von speziellen Funktionen. Diese können vom angeschlossenen Computer zumeist als 'Escape'-Sequenzen aufgerufen werden; sie werden vom Terminal selbständig ausgeführt. Das entlastet den Computer erheblich, Arbeitsgeschwindigkeit und Bedienungskomfort steigen. — Welche Möglichkeiten der c't-Terminal-Computer sonst noch bietet, und wie sein Betriebsprogramm funktioniert, erfahren Sie in dieser Beschreibung.

Ein Assembler-Listing des Betriebsprogramms liegt übrigens nicht vor, denn bislang war kein Assembler für den erweiterten Befehlssatz des 6511-Prozessors verfügbar. Das Programm wurde deshalb in Maschinencode geschrieben. Wir haben uns aus naheliegenden Gründen entschlossen, auf den Abdruck eines Hexdump zu verzichten. Dennoch sollen die Grundzüge des Programms und ein paar wesentliche Details hier erläutert werden. Wer die Daten von Hand eintippen möchte, kann unter Einsendung eines frankierten Rückumschlags das Hexdump anfordern. Betriebsprogramm und Zeichengenerator sind natürlich im c't-Software-Service auf EPROMs erhältlich.

Im Vergleich zum 6502 kennt der 6511 vier zusätzliche Befehle, von denen im Terminal-Programm häufig Gebrauch gemacht wird. Es sind die Befehle 'Reset Memory Bit' (RMB), 'Set Memory Bit' (SMB), 'Branch On Bit Reset' (BBR) und 'Branch On Bit Set' (BBS). Tabelle 1 zeigt die entsprechenden Opcodes. Alle vier Befehle arbeiten mit Zero-Page-Adressierung. Zu beachten ist, daß die Bits 4...6 der Opcodes zum

Ausdruck bringen, auf welches Datenbit der Befehl bezogen ist. Die Branch-Befehle bilden Drei-Byte-Instruktionen, wobei das zweite Byte die Zero-Page-Adresse und das dritte, wie beim 6502 gewohnt, den Branch-Offset angibt.

Arbeitsteilung

Schon im ersten Teil dieser Projektbeschreibung wurde erwähnt, daß sich beim c't-Terminal der Single-Chip-Computer 6511 und ein Video-Controller 6545 die Arbeit teilen: Der Computer bedient die verschiedenen Schnittstellen; die Daten schreibt er in einen Speicherbereich, dessen Inhalt 'rein zufäl-

RMB (Reset Memory Bit)

```
#0 #1 #2 #3 #4 #5 #6 #7
07 17 27 37 47 57 67 77
```

SMB (Set Memory Bit)

```
#0 #1 #2 #3 #4 #5 #6 #7
87 97 A7 B7 D7 C7 E7 F7
```

BBR (Branch On Bit Reset)

```
#0 #1 #2 #3 #4 #5 #6 #7
0F 1F 2F 3F 4F 5F 6F 7F
```

BBS (Branch On Bit Set)

```
#0 #1 #2 #3 #4 #5 #6 #7
8F 9F AF BF CF DF EF FF
```

Tabelle 1.
Befehlssatz-Erweiterungen des 6511

lig' auch auf einem Monitor sichtbar gemacht werden kann. Der Controller sorgt dafür, daß ein Teil des Speichers im 50-Hz-Takt ausgelesen wird und bildet die Zeilen und Bild-Synchronisationssignale.

Die Zeichen im gemeinsamen Arbeitsspeicher weisen ASCII-Format auf. Ihre Umwandlung in einzelne Bildpunkt-Impulse erfolgt auf dem Umweg über ein EPROM, an dessen Adreßleitungen die aus dem Bildspeicher gelesenen Daten als 'Spalten-Adresse' und die vom Controller gebildete 'Zeilen-Adresse' anliegen. Für die einzelnen Bildpunkte sind im EPROM, dem Zeichengenerator, je Zeichen 16x8Bit reserviert. Davon werden allerdings maximal 13x8 verwendet, wenn das Terminal im 64-Zeichen-Betrieb arbeitet. Die höchstwertigen Adreßleitungen (bei Verwendung eines 4-KByte-EPROMs nur eine) sind mit der Attribut-Schaltung verbunden, beziehungsweise herausgeführt, so daß die Wahl zwischen verschiedenen Zeichensätzen besteht.

Video-Controller

Der Controller besitzt eine Reihe von Registern (Tabelle 2), durch deren Inhalt sich alle Bild-Parameter einstellen lassen. In die meisten dieser Register kann man nur schreiben. Lesen könnte die CPU nur fünf davon; im Terminal-Programm wird davon jedoch kein Gebrauch gemacht. Der Schreibvorgang läuft auf eine etwas umständliche Weise ab: Die CPU legt zunächst in dem Adreßregister des Controllers

ein Datenbyte ab. Dessen Wert gibt an, welches Controller-Register anschließend mit dem Datenbus verbunden werden soll. Der Vorteil dieses Verfahrens besteht darin, daß der Controller-Chip nur zwei Adressen belegt: Das Adreßregister erscheint beim c't-Terminal an der Adresse 800H, jedes ausgewählte Register an der Adresse 801H.

Reg.-Nr.	Adreß-Register	Funktion
R0	0000	Hor. Total Char.
R1	0001	Hor. Displ. Char.
R2	0010	Hor. Sync. Pos.
R3	0011	Y Sync, H Sync Widths
R4	0100	Vert. Total Rows
R5	0101	Vert. Total Adjust
R6	0110	Vert. Displ. Rows
R7	0111	Vert. Sync. Pos.
R8	0100	Mode Control
R9	0101	Scan Line
R10	0101	Cursor Start Line
R11	0101	Cursor End Line
R12	0110	Start Address(H)
R13	0110	Start Address(L)
R14	0111	Cursor Addr.(H)
R15	0111	Cursor Addr.(L)
R16	1000	Light Pen Reg.(H)
R17	1001	Light Pen Reg.(L)

Tabelle 2. 6545-Register

Die Programmierung des Controller-Chips kann hier nicht erschöpfend behandelt werden. Wer sich dafür näher interessiert, sollte sich das Rockwell-Datenblatt 29000 D67 bestellen. Für das Verständnis der folgenden Beschreibung kommt es lediglich darauf an, zu erkennen, daß Bildformat, Bildfangadresse und Cursor-Position von der CPU festgelegt werden, indem diese die entsprechenden Daten in die Controller-Register schreibt. Der Video-Controller bildet dann fortwährend alle nötigen Signale zum Auslesen des Bild-

speichers. Er liefert auch das Cursor-Signal, wann immer die betreffende Speicheradresse 'dran ist'. Hardwareseitig werden diese Impulse zum Invertieren des Video-Signals verwendet, so daß der Cursor auf dem Bildschirm in der gewünschten Form erscheint. Auch das Blinken des Cursors besorgt der Controller — programmierbar ist dies durch die Datenbits 5 und 6 im Register R10.

Speicher-'Walze'

Beim c't-Terminal-Computer steht ein Bildspeicher von 4 KByte zur Verfügung. Auf dem Bildschirm erscheint lediglich ein Ausschnitt von maximal 25x80 = 2000 Zeichen. Durch Vorgabe der Bildanfangsadresse an den Controller legt die CPU fest, welcher Teil des Speicherinhalts abgebildet wird. Die Bildanfangsadresse ist die Adresse des Zeichens, das in der linken oberen Ecke zu sehen ist. Der Controller gibt die Adressen in binärer Folge aus und zählt dabei die vorgegebene Anzahl von Reihen und Zeilen hoch. Je nach Anfangsadresse kann dabei auch die höchste Speicheradresse überschritten werden: Der Controller beginnt dann wieder bei 000, so daß Speicherende und -anfang miteinander verbunden erscheinen. Man spricht deshalb auch von einem 'ring-' oder 'walzenförmigen' Speicher.

Das 'Scrollen' des Bildinhalts ist beim c't-Terminal in Auf- und Abwärtsrichtung möglich. Dabei wird nicht etwa der gesamte Bildspeicher blockweise umgeladen, sondern die CPU gibt dem Controller eine neue Bildanfangsadresse an, die um die Anzahl der Zeichen in einer

Zeile über oder unter der bisherigen liegt. Das Verfahren erfordert nur wenige Arbeitsschritte und läuft deshalb sehr schnell ab. Im Zusammenhang mit der Besprechung der seriellen Schnittstelle wird sich zeigen, welche Bedeutung die schnelle Abwicklung solcher Aufgaben hat.

Alle wichtigen Bilddaten — auch die Anfangsadresse und vor allem die aktuelle Cursor-Position — werden in Zero-Page-Zellen (Tabelle 3) regi-

Z.P.-Adr.	Name	Funktion
40	KMODE	Input-Status
41	IFMODE	Interface-Status
42	HORDIS	Zeichen pro Zeile
43	BMODE	Betriebsstatus
4A	STARTL	Bildanfang
4B	STARTR	
4C	CURL	Cursor-Adresse
4D	CURH	
4E	VCURH	Scheinbare C.-Adr.
4F	CROW	Cursor-Spalte
50	CLINE	Cursor-Zeile
51	LINES	Gesamt-Zeile
52	CHARSL	Gesamt-Zeichen
53	CHARSH	
54	ENDL	Bildende (+ 1)
55	ENDH	
56	ININD	Empfangspufferzeiger
57	OUTIND	Empfangspufferzeiger
58	ACTEMP	Akku-Puffer
59	XTEMP	X-Reg.-Puffer
5A	POINTL	Ausgabezeiger
5B	POINTH	
5C	COUNTA	Zähler
5D	SYNC	Sync.-Position
5E	COUNTB	Zähler
5F	CHAR	Empfangenes Zeichen
60	INKEY	Tastatur-Puffer
61	YTEMP	Y-Reg.-Puffer
62	EPL	Eff. Endadresse
63	EPH	
64	TOL	Zeiger f. Blockmove
65	TOH	
66	ID	Blocknummer (Kassette)
70...DF		Serieller Empfang
E0...FF		Reserv. für Stack

Tabelle 3. Belegung der Zero Page

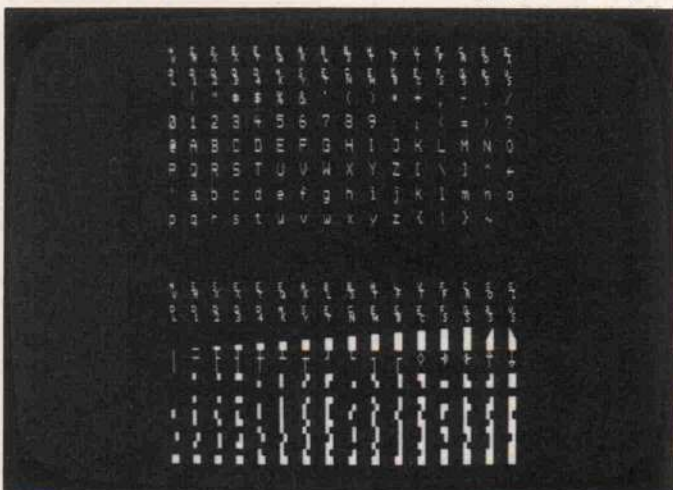


Bild 1. ASCII- und Blockgrafik-Zeichensätze im 4-KByte-EPROM

striert. Es gibt also eine 'doppelte Buchführung'. Die von der CPU gespeicherten Daten werden stets auf dem neuesten Stand gehalten und gelten deshalb als verbindlich. In der INPUT-Routine (Bild 2) aktualisiert die CPU die Controller-Register, wenn keine dringendere Aufgabe anliegt.

Das Prinzip der Bildspeicher-'Walze' — so günstig es sich auf die Arbeitsgeschwindigkeit auswirkt — bringt erhebliche Programmierungsprobleme mit sich:

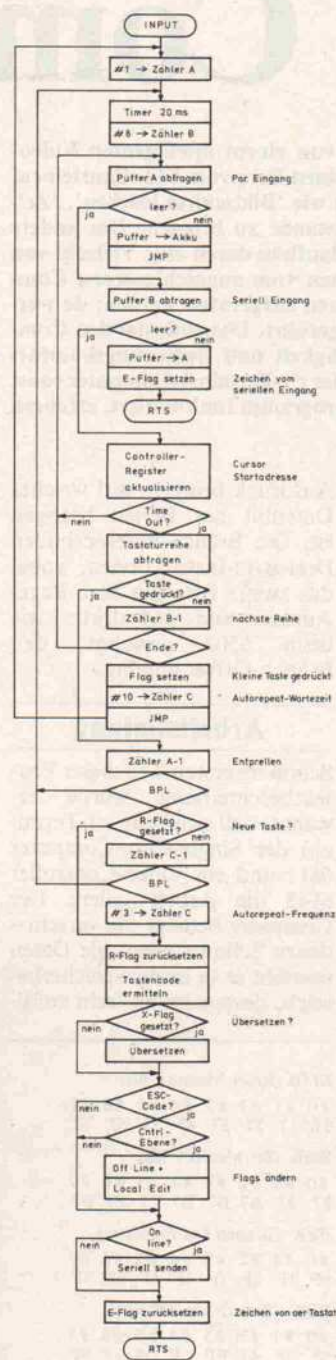


Bild 2. Schematisches Flußdiagramm der INPUT-Routine

1. Die Bildanfangsadresse kann höher sein als die Endadresse. Es läßt sich deshalb nicht durch einen schlichten Adressenvergleich feststellen, ob beispielsweise der Cursor noch im abgebildeten Speicherbereich ist. Diese Prüfung ist aber regelmäßig erforderlich, denn von ihrem Ergebnis hängt es ab, ob 'gescrollt' werden muß.

2. Der Video-Controller adressiert den Bildspeicher mit einer 14-Bit-Adresse, wovon aber lediglich 12 Bit verwendet werden. Im Adreßraum der CPU taucht der Bildspeicher im Bereich von 1000H...2000H auf. In bestimmten Situationen unterscheidet sich die für den Controller geltende Cursor-Adresse von der für die CPU geltenden. Bild 3 zeigt ein Beispiel in schematischer Darstellung. In dem skizzierten Fall arbeitet der Controller auf einen 'scheinbaren' Adreßbereich. Weil die höchstwertigen Adreßbits nicht verwendet werden, unterscheiden sich effektive und logische Adresse. Damit dennoch der Cursor am richtigen Platz abgebildet wird, muß die CPU dem Controller die scheinbare Cursor-Adresse vorgeben.

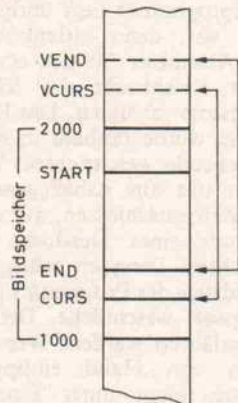


Bild 3. Scheinbarer Adreßbereich des Video-Controllers

Aus diesen beiden Gründen führt die CPU auf der Zero Page ein zweites höherwertiges Byte der Cursoradresse, das in den betreffenden Fall den scheinbaren Wert aufweist. Auch die Endadresse (ENDL/ENDH) kann einen scheinbaren Wert annehmen.

3. Arbeitet das Terminal im 80-Zeichen-Betrieb, dann gibt es keine lückenlose Bild-'Walze': Der Bildspeicher faßt in diesem Fall nämlich 51 Zeilen und 16 Zeichen. In Zeile 52 würde der Inhalt der ersten Zeile — um 16 Zeichen nach rechts verschob-

ben — auf dem Bildschirm erscheinen. Damit das nicht geschieht, löscht die CPU automatisch die folgende Zeile, wenn man in Zeile 51 ein 'Line Feed'-Kommando (AH) gibt. Die Leerzeile gilt jetzt als Zeile 51. Die bisherige erste Zeile — nun teilweise gelöscht — ist dann natürlich nicht mehr vorhanden. Das Abwärts-Scrollen stoppt, sobald die bisherige zweite Zeile oben auf dem Bild erscheint. Auf- und Abwärts-Scrollen werden übrigens automatisch ausgeführt, wenn der Cursor den Bildrand erreicht.

4. In bestimmten Fällen — beispielsweise beim Entwurf eines Briefes auf dem c't-Terminal-Computer — kann es wünschenswert sein, das Überschreiben des Speicheranfangs zu verhindern. Es sind dafür spezielle Steuerbefehle vorgesehen: 'Cursor Down' (0CH) bewirkt lediglich ein Scrollen bis zum Ende des Bildspeichers und wird dann nicht mehr akzeptiert. 'Cursor Up' (0BH) wird entsprechend nur bis zum Anfang des Bildspeichers ausgeführt. 'New Line' (1FH) bewirkt 'Carriage Return' und 'Cursor down' — im Gegensatz zum 'Carriage Return'-Befehl mit automatischem 'Line Feed', das man über eine Escape-Sequenz einstellen kann.

5. Verschiedene Editier-Kommandos — beispielsweise 'Zeile einfügen' oder 'Löschen bis Seitenende' — und einige Ausgabe-Befehle — etwa 'Seite senden' oder 'Ab Cursor bis Seitenende drucken' — wirken nicht nur auf den abgebildeten Teil, sondern auf den gesamten Bildspeicher. Da Seitenanfang und -ende die unterschiedlichsten Adressen annehmen können, muß die CPU regelmäßig 'Buch führen', um die betreffenden Kommandos jederzeit ausführen zu können.

Die CPU erledigt die Verwaltungsaufgaben mit Hilfe zweier Zeiger auf der Zero Page, in denen die aktuelle Cursor-Position als Spalten- und Zeilenangabe geführt wird. Die Zeilenangabe ist dabei auf den gesamten Bildspeicher bezogen. Die CPU kann auf diese Weise jederzeit berechnen, bis zu welcher Zeile abwärts 'gescrollt' werden darf.

Datenverkehr

Nach dem Einschalten befindet

sich das Terminal zunächst im Off-Line-Betrieb. Das heißt, es sendet keine Daten über die serielle Schnittstelle. Es ist nach jedem Reset ein Sieben-Bit-Übertragungsformat ohne Parity-Prüfung bei 4800 Baud eingestellt. Mit ESCn kann der Anwender ein Menü aufrufen, das die Einstellung aller gängigen Formate erlaubt. ESCb bringt ein Auswahl-Menü von 12 Baudraten auf den Bildschirm. Dank des zeitoptimierten Programms erzielt das Terminal eine maximale Übertragungsgeschwindigkeit von 19200 Baud mit relativ guter Genauigkeit.

ESCX schaltet das Terminal in den On-Line-Betrieb um. Es wird dann jedes von der Tastatur empfangene Zeichen zum angeschlossenen Computer gesendet. Auch ESC wird in dieser Betriebsart gesendet und führt nicht zum Eintritt in eine Escape-Sequenz. Anders ist das, wenn das ESC-Steuerzeichen über die serielle Schnittstelle empfangen wird. Wer die Terminal-Version B (mit Low-Cost-Tastatur auf der Platine) verwendet, kann allerdings durch Cntrl./ESC die Verbindung unterbrechen und eine Escape-Sequenz direkt aufrufen.

Der serielle Eingang wird, ebenso wie die Parallelschnittstelle an Port B, in einer Interrupt-Routine bedient. Dabei hat die Parallelschnittstelle, die für den Anschluß einer Tastatur mit eigenem Decoder gedacht ist, die höhere Priorität. Geht hier ein Zeichen ein, so legt der Computer dies in einem Puffer (IN-KEY) ab und setzt eine Flag. Wird am seriellen Eingang ein Zeichen empfangen, so gelangt dies in der Interrupt-Routine zunächst in einen Empfangspuffer auf der Zero Page, den der Computer nach dem FIFO-Konzept ('First In — First Out') verwaltet. Dieser Empfangspuffer umfaßt 112 Speicherplätze.

Ein solcher Puffer ist angebracht, weil bei höchster Übertragungsgeschwindigkeit nur etwa 0,4 msec zur Verfügung stehen, um jedes empfangene Zeichen zu verarbeiten. Das Ablegen eines Zeichens im Bildspeicher, zusammen mit dem jedesmal erforderlichen Verwaltungsanteil, braucht zwar deutlich weniger Zeit. Ruft der Computer aber eine Escape-Sequenz auf, die mit größerem

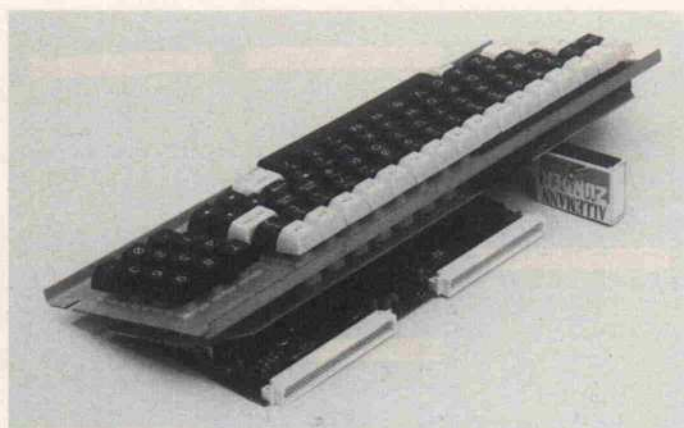


Bild 4. Die Terminal-Version A im Format einer halben Doppel-Europakarte läßt sich zusammen mit einem Flachtrafo unter einer ProfiTastatur unterbringen

Rechenaufwand verbunden ist, so wird eine gewisse Zeitreserve benötigt. Für den Notfall ist außerdem ein Busy-Signal vorgesehen, das freilich nicht der V-24-Norm entspricht: Beim Überlauf des Empfangspuffers legt das Terminal die Leitung PA1 auf log. 0. Wird dieser Anschluß von außen auf Massepegel gelegt, so unterbricht das Terminal das Senden über die serielle Schnittstelle. Auf diese Weise kann auch der angeschlossene Computer ein Busy-Signal abgeben. Es ist aber darauf zu achten, daß eine Open-Collector-Schaltung dafür verwendet werden muß.

Sowohl der Puffer der Parallelschnittstelle als auch der FIFO-Speicher werden in der INPUT-Routine geleert, die den Zentralpunkt des Terminal-Programms bildet. Nach jeder Befehlsausführung kehrt die CPU dorthin zurück. Sind beide Puffer leer, frischt sie die Controller-Register für die Bildanfänger- und die Cursor-Adresse auf. Auch die Abfrage der Tastatur bei Version B wird in dieser Routine erledigt.

Tastatur

Die in der Version B vorhandene Tastatur wird vom 6511-Computer abgefragt. Zur Entprellung und zum Erzeugen von Autorepeat ist einer der 16-bit-Timer eingesetzt. Der Computer fragt jeweils eine Matrixreihe ab, wenn alle übrigen Aufgaben in der INPUT-Routine erledigt sind. Die gesamte Tastaturabfrage wird alle 50 msec wiederholt.

Wird eine Taste gedrückt, so ermittelt die CPU nach einer Entprellzeit von 50 msec einen Ta-

stenwert, der als Index zur Adressierung einer Tastenliste dient. Es sind für die drei Tastaturebenen getrennte Listen vorhanden, so daß jeder Taste sowohl auf der unteren, als auch auf der oberen und der Control-Ebene ein beliebiger ASCII-Code zugeordnet werden kann. Wahlweise (ESC ; — ESC :) wird der Code anhand einer Tabelle übersetzt. Diese Übersetzung ist zum Beispiel bei Verwendung der deutschen Umlaute notwendig, da das ASCII-Schema der Zuordnung von Groß- und Kleinschrift hier durchbrochen wird. Übrigens werden auch die an der Parallelschnittstelle empfangenen Daten der Übersetzung unterzogen.

Die Versionen A und B unterscheiden sich in der Software nicht. Das heißt, auch bei Version A, die nicht mit eigener Tastatur ausgestattet ist, führt die CPU eine Matrixabfrage aus. An den Steckerleisten X3 und X4 kann man also ohne jede Änderung eine Tastatur mit offener 7x8-Matrix anschließen, wenn die Tastenbelegung dem in c't 12/83 veröffentlichten Schema entspricht. Mit entsprechender EPROM-Änderung läßt sich auch eine Tastatur mit 8x9-Matrix verwenden. Die Abfrageroutine ist darauf bereits eingerichtet; lediglich die Codelisten müssen noch angepaßt werden. Der dafür benötigte Speicherplatz ist im EPROM freigelassen.

Screen Editing

Der c't-Terminal-Computer bietet bequeme Möglichkeiten zum Editieren auf dem Bildschirm. Diese können sowohl als Escape-Sequenzen (Bild 5)

c't-Terminal Escape-Sequenzen		Editieren		Input	
Cursor-Positionierung					
Rel. Positionierung		Seite löschen	+ (2B)	Send Line	
↑	BS (08)	Z	(5A)	Send Page	
↑	FF (0C)	Zeilenende l.	Y (5B)	On Line	
↑	VT (0A)	Zeilenende r.	T (54)	Off Line	
↑	LF (0B)	Zeichen einf.	O (51)	Local Edit	
Load Cursor	= (3D)	Zeichen ausf.	W (57)	Duplex Edit	
Read Cursor	? (3F)	Zeile einf.	I (69)	Read Curs. Char	
		Zeile ausf.	X (52)	Port On	
Attribute					
bis Zeilende	BP (20)	Pos. merken	= (60)	Port Off	
halbhell	(21)	Zeichentausch	x (78)	Print Line	
invers	~ (22)	Keyboard			
doppelt breit	# (23)	Alpha Lock	a (61)	Print Page	
blinken	= (24)	Caps Lock	z (7A)	Print + Curs.	
Grafik	* (25)	Auto LF On	G (+7)	Print Curs. +	
deutsch/Graf. 2 &	& (26)	Auto LF Off	H (+8)	Cursor-Form	
Attr. löschen	~ (27)	Codelliste A	(3A)	Linie blinkend	~ (71)
Bild Invers	y (79)	Codelliste B	(3B)	stehend	~ (77)
Bildformat					
30x25	0 (30)	Kassetten-Interface			
64x20	1 (31)	Save "n"	s (73)	Block blinkend	~ (74)
Hor.Sync. H	.. (2C, 3E)	Get "n"	g (67)	stehend	~ (72)
				Wicht sichtbar	= (65)

Bild 5. Escape-Sequenzen und die dazugehörigen Befehlscodes

wie auch durch Steuerbefehle (Bild 6) aufgerufen werden. Die Verwendung der Steuerbefehle im Off-Line-Betrieb bietet dabei den Vorteil, daß die Autorepeat-Funktion ausgenutzt werden kann.

Funktionen wie 'Leerzeile einfügen' oder 'Zeile löschen', die jeweils auf den gesamten Bildspeicher wirken, erleichtern den Entwurf von Texten und Programmen auf dem Bildschirm. Mit 'Send Line' oder 'Send Page' lassen sich die Daten aus dem Bildspeicher zum angeschlossenen Computer übertragen. Leerzeichen am Ende einer Zeile werden dabei nicht gesendet. Jede Zeile wird mit 'Carriage Return' (QDH) abgeschlossen.

Eine Besonderheit stellt die Funktion 'Zeichentausch' dar. Diese ermöglicht es, einzelne Zeichen oder ganze Absätze ge-

geneinander auszutauschen. Das funktioniert so: Der Cursor wird an die Position des einen Zeichens beziehungsweise den Anfang eines Absatzes gebracht. ESCm oder Cntrl./B bewirkt dann, daß das Terminal diese Position speichert. Anschließend fährt man den Cursor zu dem auszutauschenden Zeichen beziehungsweise an den Anfang des auszutauschenden Textblocks und gibt mit ESCx oder Cntrl./C den Befehl zum Platzwechsel. Bei jeder Wiederholung (Autorepeat) tauschen dann zwei weitere Zeichen die Plätze.

Auch die Möglichkeiten der Cursor-Positionierung sind sehr komfortabel: Neben der absoluten Positionierung, wie sie zum Beispiel 'WordStar' fordert (ESC =, gefolgt von Reihe/Spalte in hexadezimaler Angabe, mit einem Offset von 20H), ist auch die relative Posi-

c't-Terminal Control Codes			
Funktion	Cntrl	Funktion	Cntrl
Testbild	@	Steuerz. nicht zeigen	T
Zeichensatz	A	Zeichen einfügen	U
Curs.-Pos. merken	B	Zeichen ausfügen	W
Austauschen	C	Zeilenende löschen	V
Signal	G	Seitenende löschen	X
Cursor +	H	Zeile einfügen	Y
Tabulator	I	Zeile ausfügen	Z
Line Feed	J	Escape	;
Cursor ↑	K	Cursor ↓	↓
Cursor →	L	Seite löschen	+
Carriage Return	M	Home	~
Steuerzeichen zeigen	R	Neue Zeile	-

Bild 6. Steuerbefehle

tionierung nach demselben Schema in jeder Richtung möglich.

Auf Kassette retten

Absolut unüblich ist es, ein Terminal mit einem eigenen Kassetteninterface auszustatten. Da bei dem c't-Terminal-Computer aber durchaus die Möglichkeit des Ausbaus zu einem Stand-Alone-Computer oder Textverarbeitungssystem besteht (wir arbeiten schon daran), wurde auch dies softwareseitig schon vorgesehen. Die Hardware ist dabei kaum der Rede wert — eine entsprechende Schaltungsergänzung enthielt der erste Teil der Projektbeschreibung.

Es wird das sehr schnelle Super-tape-Aufzeichnungsverfahren (1) benutzt, das für den COBOLD-Computer unserer Schwesterzeitschrift elrad entwickelt worden ist. Damit gelingt es, den gesamten Bildspeicherinhalt und die wichtigsten Bild-Parameter in weniger als sieben Sekunden auf eine Kompakt-Kassette zu retten oder von dieser zu lesen. Man kann also Formbriefe, BASIC-Programme, Bildschirmmasken und sonstwas auf Dauer aufbewahren.

Der Befehl zum Retten auf die Kassette lautet ESCs, gefolgt von einer Ein-Byte-Identifikationsnummer. Die Nummer kann aus beliebigen hexadezimalen Ziffern bestehen (0...9 und A...F). Die Eingabe wird, nach dem Start des Recorders, mit RETURN abgeschlossen; als Identifikationsnummer speichert der Computer stets die beiden zuletzt eingegebenen Ziffern.

Um die Daten vom Band zu lesen, geben Sie bitte ESCg und die betreffende Nummer an. Wenn sich auf dem Band weitere Files befinden, zeigt der Terminal-Computer deren Nummern auf dem Bildschirm an. Findet er den gesuchten Datenblock, so lädt er diesen in den Bildspeicher und stellt anschließend die bei der Aufnahme vorhandenen Bild-Parameter sowie die Betriebsart-Einstellungen wieder her.

Hard-Copy

Eine weitere Möglichkeit, den Bildschirminhalt zu 'verewigen', bietet der Centronics-Ausgang, an den sich jeder

Drucker mit entsprechender Schnittstelle anschließen läßt. Das Terminal überträgt alle Daten im Sieben-Bit-Code — Attribute werden nicht übertragen. Allerdings sind in die Ausgaberroutine einige NOPs eingefügt, so daß die Anpassung an spezielle Druckertypen nachträglich in einem Unterprogramm vorgenommen werden kann.

ESCP bewirkt die Ausgabe des gesamten Bildspeicherinhalts an den Drucker. Auch Leerzeilen werden übertragen. Papier sparen kann man mit ESCN (Ausgabe von Anfang bis Cursor-Position) oder ESC0 (Ausgabe von Cursor-Position bis Ende). Nach Ausgabe einer jeden Zeile sendet das Terminal 'Carriage Return' und 'Line Feed' — ist jedoch 'Auto-Line-Feed' eingestellt, so wird jede Zeile nur mit 'Carriage Return' beendet; der Drucker sollte dann ebenfalls auf automatischen Zeilenvorschub eingestellt sein.

Programmstruktur

Das Terminal-Betriebsprogramm besteht aus drei Blöcken, deren Zusammenhang in Bild 7 skizziert ist:

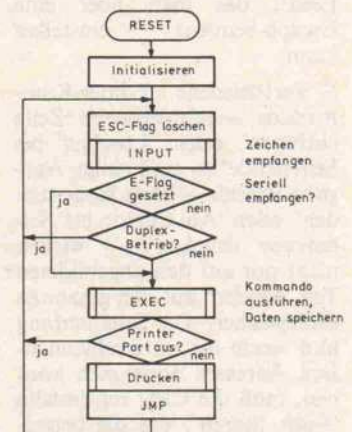


Bild 7. Grobstruktur des Terminal-Programms

Die Initialisierungsphase durchläuft die CPU nach jedem Reset, also auch nach dem Einschalten (Power-On-Reset). Sie initialisiert den Stackpointer (der Stack liegt beim 6511 übrigens auf der Zero Page), legt verschiedene Daten zur Betriebsartenauswahl in den dafür vorgesehenen Speicherzellen ab, programmiert den Timer A als Baudratengenerator für die serielle Schnittstelle und kopiert eine Liste von Daten in die Register des Video-Controllers.

Anschließend füllt sie den gesamten Bildspeicher mit dem SP-Code (20) und kopiert die Begrüßungsmeldung in die erste Bildzeile. Wenn diese Vorgänge abgeschlossen sind, setzt die CPU die Interrupt-Flag im Statusregister zurück, so daß beim Empfang von Daten die Interrupt-Serviceroutine aufgerufen wird.

Damit ist das Terminal betriebsbereit. Die CPU befindet sich nun in der INPUT-Routine; sie bleibt dort in einer Warteschleife, in deren Verlauf sie regelmäßig die Empfangspuffer überprüft, die Controller-Register auffrischt und die Tastaturmatrix abfragt. Währenddessen kann jederzeit ein Interrupt ausgelöst werden, wenn am seriellen Eingang oder an der Parallelschnittstelle (Port B) Daten eintreffen. Die serielle Schnittstelle gibt bei entsprechender Programmierung einen Interrupt Request (IRQ) ab, sobald sich ein Datenwort im Eingangsregister befindet. Der Dateneingang an der Parallelschnittstelle muß durch einen positiven Strobe-Impuls an PA0 oder durch einen negativen Strobe an PA2 angezeigt werden; beide Eingänge sind als Flankendetektoren programmiert. In der Interrupt-Routine legt die CPU das empfangene Byte im betreffenden Empfangspuffer ab und kehrt wieder in das unterbrochene Programm zurück. Beim nächsten Durchlaufen der INPUT-

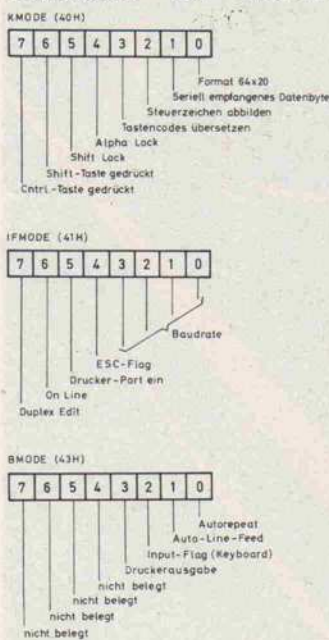


Bild 8. Die verschiedenen Betriebszustände des Terminals schlagen sich in 'Soft-Flags' nieder.

Schleife entnimmt sie das Datenbyte dem Puffer und verläßt die INPUT-Routine.

Zunächst werden alle empfangenen Daten gleich behandelt, unabhängig davon, ob sie von der Tastatur oder über die serielle Schnittstelle eingegangen sind. Die CPU setzt jedoch ein Flag-Bit in der Speicherzelle KMODE, wenn ein Datenbyte von der seriellen Schnittstelle kommt. Befindet sie sich im On-Line-Betrieb, so werden andererseits die von der Tastatur empfangenen Daten noch in der INPUT-Routine zum Rechner gesendet.

Der dritte Abschnitt mit Namen EXEC dient zur Verarbeitung der empfangenen Daten. Die CPU verzweigt zur Escape-Bearbeitung, wenn zuvor das ESC-Steuerzeichen empfangen worden war. In diesem Fall hätte sie nämlich ein Flag-Bit in der Speicherzelle IFMODE gesetzt, das zu Beginn der EXEC-Routine getestet wird. Ist das Bit nicht gesetzt, so prüft die CPU, ob das empfangene Zeichen einen Steuerbefehl beinhaltet (Codes 00...1F). Andernfalls wird das Zeichen an der Cursor-Position im Bildspeicher abgelegt, der Cursor weitergerückt, gegebenenfalls die Zeilennummer um eins erhöht und, falls nötig, das Bild 'gescrollt'.

Die Bearbeitung der Steuerzeichen und die Ausführung der Escape-Sequenzen erfolgen auf identische Weise: Der im Akku enthaltene Tastencode wird als Index zur Adressierung von Sprungtabellen verwendet. Die Sprungadresse legt die CPU auf dem Stack ab. Sie trifft dann auf einen RTS-Befehl und springt infolgedessen zu dem Programm, dessen Adresse sie zuvor der Tabelle entnommen hatte. Dieses Programm endet wiederum mit RTS, was die CPU in die nächsthöhere Programmebene zurückbringt, denn EXEC wurde gleichfalls als Subroutine aufgerufen.

Der Aufbau des Terminal-Programms macht es relativ leicht möglich, Modifikationen und Erweiterungen vorzunehmen. Zusätzliche Unterprogramme können durch entsprechende Ergänzung der Sprungtabellen angefügt werden. Änderung der Steuer- und Escape-Codes sind einfach durch Vertauschen der Sprungadressen zu bewerkstelligen.

c't-Terminal Betriebsprogramm

Listen:

- 9000: Tastencodes (untere Ebene)
- 9048: Tastencodes (obere Ebene)
- 9090: Tastencodes (Steuerzeichen)
- 90E8: Tastencode-Übersetzung
- 9158: Daten für Video-Initialisierung
- 9178: Zeichenanzahl bei 80x25/64x20
- 9180: Subroutinen-Adr. (L) für Steuerbefehle
- 91A0: Subroutinen-Adr. (H) für Steuerbefehle
- 91C0: Subroutinen-Adr. (L) für Escape-Sequenzen
- 9240: Subroutinen-Adr. (H) für Escape-Sequenzen
- 92C0: Timer-Presets für versch. Baudraten
- 92D8: Textfiles (Meldung, Menüs)
- 9499: Ende

Wichtige Unterprogramme

- 9600: INPUT-Routine: fragt Puffer der parallelen und der seriellen Schnittstellen ab, aktualisiert Bildanfangs- und Cursoradressen, fragt Tastaturmatrix ab, sendet von Tastatur empfangenes Zeichen (wenn On Line). Rückkehr mit empfangenem Zeichen im Akku und Bit #1 KMODE = 1, wenn das Zeichen aus dem Puffer der seriellen Schnittstelle kommt.
- 970E: Interrupt-Routine: Bedient parallele oder serielle Schnittstelle, je nach Anforderung. Legt empfangenes Zeichen im entsprechenden Puffer ab.
- 976C: Stellt Timer A auf gewählte Baudrate (Bits #0...3 IFMODE) ein.
- 977C: Stellt Video-Controller auf gewähltes Format (Bit #0 KMODE) ein.
- 97C2: Rückt Cursor um eine Stelle nach rechts.
- 97DA: Rückt Cursor um eine Stelle nach links.
- 9823: Bewirkt Carriage Return und wahlweise Line Feed (Bit #1 BMODE).
- 984D: Löscht Zeile ab Cursorposition
- 9865: Bewegt Cursor um eine Zeile aufwärts; scrollt gegebenenfalls Bild abwärts.
- 9889: Bewegt Cursor um eine Zeile abwärts; scrollt gegebenenfalls aufwärts.
- 98E9: EXEC: Bereitet Ausführung von Escape-Sequenzen und Steuerbefehlen vor, speichert Zeichen im Video-RAM.
- 991B: RESET-Adresse, Terminal-Initialisierung.
- 997F: Ausgabe eines Zeichens an den Drucker/die serielle Schnittstelle (Bit #3 BMODE).

- 99AB: Tabulator mit festen Positionen (alle 8 Spalten).
- 99B6: Empfängt Bildattribut-Befehl und setzt Attribute.
- 99D6: Empfängt Cursor-Position und positioniert Cursor.
- 9A17: Sendet Cursor-Position über serielle Schnittstelle.
- 9A49 ff.: Setzen Betriebsart-Flags in KMODE, IFMODE und BMODE.
- 9AD6: Gibt Baudraten-Menü aus und stellt Baudrate neu ein.
- 9B0F: Empfängt relative Cursor-Position und setzt Cursor-Adresse.
- 9B25: Überschreibt Zeichen unter dem Cursor von rechts mit dem Rest der Zeile.
- 9B4A: Fügt an der Cursorposition Leerzeichen ein; rückt Rest der Zeile nach rechts.
- 9BBE: Gibt Cursorzeile an den Drucker/die serielle Schnittstelle aus. Unterdrückt abschließende Leerzeichen, fügt Carriage Return und bei Drucker-Ausgabe wahlweise Line Feed an.
- 9BD6: Gibt Bildspeicherinhalt bis Cursorposition an Drucker aus.
- 9C55: Fügt vor der Cursorzeile eine Leerzeile ein.
- 9C8C: Überschreibt Cursorzeile; Rest des Bildspeichers rückt auf.
- 9CCF: Setzt Cursor an Bildspeicheranfang ('Home').
- 9CDD: Löscht Bildspeicher ab Cursor mit 'SP' (20H).
- 9D06: Zeichenaustausch
- 9D46: Empfängt Zeichen und wandelt in Hex-Byte um.
- 9D65: Gibt Hex-Byte in zwei Zeichen auf dem Bildschirm aus.
- 9DB5: Empfängt Identifikationsnummer und rettet Bildspeicher und Zero-Page-Variablen 40H...66 auf Kassette.
- 9E50: Empfängt Identifikationsnummer und lädt Datenblock von der Kassette.
- 9EDE: Gibt L-Impuls für akustisches Signal an PA1 aus.
- 9EEE: Bringt zwei Zeichensätze auf den Bildschirm.
- 9F36: Gibt gesamten Bildspeicher an den Drucker/die serielle Schnittstelle (Bit #3 BMODE) aus.
- 9F60: Gibt Bildspeicherinhalt von Cursorposition bis Ende an den Druckerport aus.
- 9FFA: NMI-Vektor (nicht verwendet)
- 9FFC: RESET-Vektor
- 9FFE: IRQ-Vektor

Tabelle 4. Betriebsprogramm-Adressen

Die anschließende PRINT-Routine ist ein 'Anhängsel' mit einem ganz speziellen Zweck: Wird der Drucker-Port eingeschaltet (ESC @), so gibt das Terminal jedes empfangene Zeichen einschließlich aller Steuersignale dort parallel aus. Dieses Feature kann für die gleichzeitige Ausgabe auf Bildschirm und Drucker genutzt werden; das Terminal erfüllt damit die Aufgabe eines Seriell/Parallel-Wandlers. Allerdings sollte man ESC @ nur dann geben, wenn tatsächlich ein Drucker angeschlossen ist. Andernfalls interpretiert das

Terminal den H-Pegel am (offenen) Eingang PA3 fälschlicherweise als Busy-Signal des Druckers und wartet vergeblich...

Literaturhinweise:

- R6511Q One-Chip-Microprozessor und R6500/13 One-Chip-Microcomputer — Rockwell-Datenblatt Nr. 29651N36
- R6545-1 CRT-Controller-Rockwell-Datenblatt Nr. 29000D67
- (1) Christian Persson: 6502/65C02 — Maschinensprache, Verlag Heinz Heise GmbH, Hannover