

MFA-MEDIENSYSTEM

Mikrocomputer- Technik

Fachpraktische Übungen
Band 4

4


MEDIENSYSTEM

VGS 





– Ihr Partner für Ausbildungssysteme

Neue Produkte im Bereich Mediensysteme

Mit Medien, Aus- und Weiterbildung begann sich die vgs bereits vor 20 Jahren mit dem Begleitbuch zur Fernsehreihe „Einführung in die Elektronik“ von Jean Pütz zu beschäftigen. Über die Jahre hinweg entstanden in Zusammenarbeit mit ARD und ZDF weitere Bücher und Bausätze aus den Bereichen Elektronik, Chemie und Biotechnologie. Inzwischen ist die vgs der Spezialist für Medienverbund auf allen Gebieten. Die Zusammenarbeit mit dem Berufsförderungszentrum Essen e.V. begann 1983, als die vgs den Zuschlag für die Produktion und Verbreitung des dort entwickelten Mikrocomputer-Ausbildungssystems MFA erhielt. Inzwischen umfaßt das Programm der vgs für den Bereich der Aus- und Weiterbildung folgende Produkte:

- **MEA** – ein Mikrocomputer-Ausbildungssystem, das in den industriellen Metall- und Elektro-Berufen sowie in den handwerklichen Elektro-Berufen im In- und Ausland eine zentrale Rolle spielt. Das vom BFZ in Essen entwickelte Grundsystem wird von der vgs ständig zukunftsorientiert weiterentwickelt, produziert und vertrieben, so daß inzwischen ca. 90 Baugruppen zur Verfügung stehen.
- **40 900 NORMCOMPUTER** – ein Computer-Lehrsystem für die Schulung im Bereich Digitaltechnik, bei dem Wert auf die Einhaltung der gültigen internationalen Digitalnorm nach DIN 40 900 gelegt wurde (in Zusammenarbeit mit dem BBZ Köln). Mit diesem, aus 8 Lehrplatten bestehendem Lehrsystem, kann die immer noch bestehende Lücke zwischen Digitaltechnik und Mikrocomputertechnik geschlossen werden. Basis des NORMCOMPUTERS ist der in der Aus- und Weiterbildung nach wie vor optimal einzusetzende Mikrocomputer 8085.
- **PC-Modelle** für die Schulung – hier bietet die vgs Anwendungsmodelle aus den Bereichen Logikanalyse, Meßtechnik und Robotik an.
- **PTQ** (steht für Produktionstechnische Qualifikation im Lernverbund). An dieser neuen Entwicklung vom BFZ Essen ist die vgs als Werkvertragsnehmer über die Konstruktion und Dokumentation sowie Fertigungsarbeiten für „Portallader-Komponenten“ beteiligt.

Die vgs liefert in das Inland (inzwischen mit Schwerpunkt neue Bundesländer) und über Vertragspartner in das Ausland an Industrie, Handwerk, Ausbildungszentren und Schulen.

Bitte fordern Sie weitere Informationen an bei:

vgs verlagsgesellschaft mbH & Co. KG
Postfach 1802 69
Breite Straße 118-120, 5000 Köln 1

Telefon 02 21/2 08 11-12
Telefax 02 21/24 57 99
Telex 888 2202 vgs d

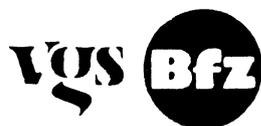


MFA-MEDIENSYSTEM

Mikrocomputer- Technik

Fachpraktische Übungen · Band 4

- Teil 1 Ausbaustufe MT
- Teil 2 Ausbaustufe PC
- Teil 3 Ausbaustufe MT & PC



MFA-Mediensystem Mikrocomputer-Technik / [Red.: F.

Derriks ...]. – Köln : vgs.

Teilw. redigiert von N. Meyer ... Teilw. hrsg. vom BFZ Essen

NE: Meyer, Norbert [Red.]; Derriks, Franz [Red.];

Berufsförderungszentrum <Essen>; Mikrocomputer-Technik

Fachpraktische Übungen.

Bd. 4. – 1. Aufl. – 1992

ISBN 3-8025-1253-7

Herausgeber: vgs verlagsgesellschaft mbH & Co. KG
Breite Str. 118-120,
5000 Köln 1

Redaktion: F. Derriks, M. Hüllweg, Fr.G. Roßmanek,
K.D. Strelow, H. Storbeck, S. Wirtgen,
R. Krenz, H. Steinmann, W. Hild,
W. Oehlert, D. Piller

C 1991 vgs verlagsgesellschaft mbH & Co. KG
Berufsförderungszentrum Essen e.V.

Diese Publikation ist urheberrechtlich
geschützt. Alle Rechte sind vorbehalten.

Verlag: vgs verlagsgesellschaft, Köln

1. Auflage 1991

Satz und Zeichnungen: BFZ Essen; Teba-Elektronik, Nenderoth

Hinweise zu den Übungen in diesem Band

Die in diesem Band enthaltenen Übungen sind nicht durchgehend, sondern einzeln paginiert, und zwar oben rechts auf jeder Seite. Dabei ist jeweils auch die Systemnummer der betreffenden Übung angegeben (z.B. Video-Interface BFZ/MFA 8.4). Hierdurch ist es möglich, den Band auseinanderzunehmen und die Übungen einzeln einzusetzen.

Dieser Band enthält folgende Übungen:

Teil 1 Ausbaustufe MT

- ASCII-Tastatur
- Video-Interface
- Schnelleres Video-Interface

Teil 2 Ausbaustufe PC

- V.24/RS232-Schnittstelle
- IBM-Terminalprogramm

Teil 3 Ausbaustufe MT und PC

- Beschreibung des MAT32K bzw. MAT32K-Terminalversion
- 64-K-RAM-Karte

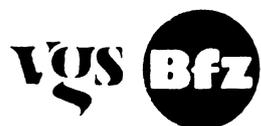
MFA-MEDIENSYSTEM

Mikrocomputer- Technik

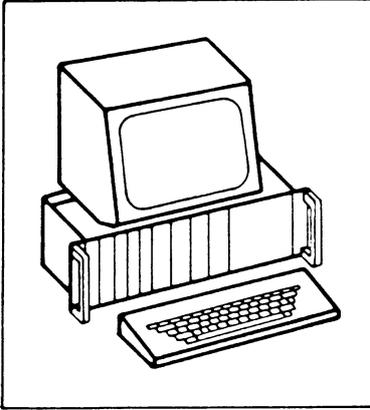
Fachpraktische Übungen · Band 4

Teil 1

Ausbaustufe MT



FACHPRAKTISCHE ÜBUNG MIKROCOMPUTER-TECHNIK



ASCII-Tastatur

BFZ/MFA 8.1.



ASCII - Tastatur

1. Einleitung

Die verwendete Tastatur besitzt 65 Tasten. Mit ihnen lassen sich folgende Funktionen erzeugen:

- der 7-Bit-Code für ein alphanumerisches Zeichen
- der 7-Bit-Code für ein Sonderzeichen
- der 7-Bit-Code für ein Steuerzeichen
- die Umschaltung von einer Tastenfunktion auf eine andere derselben Taste
- Steuerfunktionen für den Datentransport zu angeschlossenen Geräten

Die Kodierung aller genannten Zeichen ist durch die ASCII-Norm festgelegt. ASCII ist die Abkürzung für American Standard Code for Information Interchange.

Alle alphanumerischen- und Sonderzeichen sind, z.B. mit einem Datensichtgerät, darstellbar. Steuerzeichen und Umschaltfunktionen sind nicht darstellbar, sie bewirken nur bestimmte Dinge, wie z.B. einen Zeilenvorschub, einen Wagenrücklauf oder die Umschaltung von Groß- auf Kleinschreibung. Eine Kurzbeschreibung der Umschaltfunktionen finden Sie im Abschnitt 3 dieser Funktionsbeschreibung.

2. Kodierung der Zeichen

Bild 1 zeigt eine Tabelle der ASCII - Zeichen, die international angewendet werden. Diese Tabelle enthält keine Umlaute. In den Spalten 0 und 1 finden Sie die Steuerzeichen. Die jedem Zeichen entsprechende Bitkombination des 7-Bit-Codes kann den Spalten b1 bis b7 entnommen werden.

Als Beispiel sei die Kodierung des Zeichens "M" erläutert. Dieses Zeichen steht in der 4. Spalte der 13. Zeile der Tabelle. Die binäre Darstellung des Zeichens ergibt sich daher zu:

1	0	0	1	1	0	1	≅	M
---	---	---	---	---	---	---	---	---

Hexadezimal dargestellt ist dies:

4		D	≅	M
---	--	---	---	---

ASCII - Tastatur

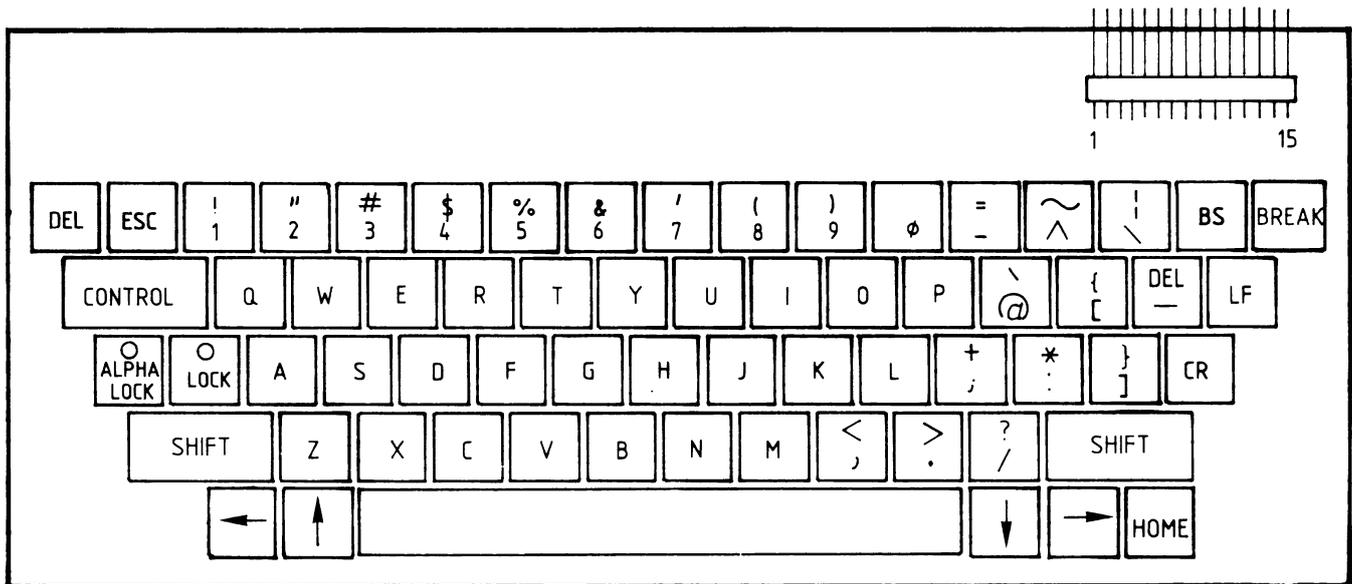
Bit	b_7	b_6	b_5	b_4	b_3	b_2	b_1	Zeile	Spalte							
									0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0	NUL	TC ₇ (DLE)	SP	0	@	P	\	p
0	0	0	0	1	1	0	0	1	TC ₁ (SOH)	DC ₁	!	1	A	Q	a	q
0	0	0	1	0	0	0	0	2	TC ₂ (STX)	DC ₂	"	2	B	R	b	r
0	0	0	1	1	1	0	0	3	TC ₃ (ETX)	DC ₃	#	3	C	S	c	s
0	1	0	0	0	0	0	0	4	TC ₄ (EOT)	DC ₄	\$	4	D	T	d	t
0	1	0	0	1	0	0	0	5	TC ₅ (ENQ)	TC ₈ (NAK)	%	5	E	U	e	u
0	1	0	1	0	0	0	0	6	TC ₆ (ACK)	TC ₉ (SYN)	&	6	F	V	f	v
0	1	0	1	1	0	0	0	7	BEL	TC ₁₀ (ETB)	'	7	G	W	g	w
1	0	0	0	0	0	0	0	8	FE ₀ (BS)	CAN	(8	H	X	h	x
1	0	0	0	1	0	0	0	9	FE ₁ (HT)	EM)	9	I	Y	i	y
1	0	0	1	0	0	0	0	A	FE ₂ (LF)	SUB	*	:	J	Z	j	z
1	0	0	1	1	0	0	0	B	FE ₃ (VT)	ESC	+	;	K	[k	{
1	1	0	0	0	0	0	0	C	FE ₄ (FF)	IS ₄ (FS)	,	<	L	\	l	
1	1	0	0	1	0	0	0	D	FE ₅ (CR)	IS ₃ (GS)	-	=	M]	m	}
1	1	0	1	0	0	0	0	E	SO	IS ₂ (RS)	.	>	N	^	n	_
1	1	0	1	1	0	0	0	F	SI	IS ₁ (US)	/	?	O	~	o	DEL

Bild 1: ASCII-Code- Tabelle, internationale Referenzversion

ASCII - Tastatur

3. Ansicht der Tastatur, Beschriftung der Tasten und hexadezimale Verschlüsselung der Tastenfunktionen

Bild 2 zeigt die Ansicht der Flach tastatur G80 - 0177 der Fa. Cherry und den ASCII - Code für die Zeichen und Funktionen jeder Taste.



7F	1B	31	32	33	34	35	36	37	38	39	30	2D	1E	1C	08	
7F	1B	21	22	23	24	25	26	27	28	29	30	3D	7E	7C	08	BREAK
7F	1B	31	32	33	34	35	36	37	38	39	30	2D	5E	5C	08	
CONTROL		11	17	05	12	14	19	15	09	0F	10	00	1B	1F	0A	
		51	57	45	52	54	59	55	49	4F	50	60	7B	7F	0A	
		71	77	65	72	74	79	75	69	6F	70	40	5B	5F	0A	
ALPHA LOCK	LOCK	01	13	04	06	07	08	0A	0B	0C	3B	3A	1D	0D	0D	
		41	53	44	46	47	48	4A	4B	4C	2B	2A	7D	0D	0D	
		61	73	64	66	67	68	6A	6B	6C	3B	3A	5D	0D	0D	
SHIFT		1A	18	03	16	02	0E	0D	2C	2E	2F	SHIFT				
		5A	58	43	56	42	4E	4D	3C	3E	3F					
		7A	78	63	76	62	6E	6D	2C	2E	2F					
08	0B	CONTROL									0A	09	0F			
08	0B	SHIFTED									0A	09	0F			
08	0B	UNSHIFTED									0A	09	0F			

Bild 2: Ansicht der Flach tastatur mit Beschriftung und Kodierung der Tasten

ASCII - Tastatur

Die Kodierung ist in hexadezimaler Form angegeben. Tasten, die nicht mit dem ASCII - Code beschriftet sind, haben Umschaltfunktionen.

Beschreibung der Umschaltfunktionen, Wirkung der Taste:

- CONTROL : Bei gleichzeitiger Betätigung dieser Taste und einer weiteren wird üblicherweise ein Steuerzeichen erzeugt (control = steuern).
- SHIFT : Bei Betätigung der SHIFT-Taste wird auf die obere Tastenfunktion (Großschreibung, Satzzeichen usw.) umgeschaltet (shift = schieben, verlagern).
- LOCK : Mit dieser Taste wird die SHIFT-Funktion festgesetzt. Ist die Taste betätigt worden, so wird dies durch eine eingebaute Leuchtdiode angezeigt. Durch abermaliges Betätigen der Taste wird die Verriegelung wieder aufgehoben (lock = festhalten).
- ALPHA LOCK : Diese Taste hat eine ähnliche Funktion wie die oben beschriebene SHIFT-LOCK-Funktion. Der Unterschied besteht darin, daß nur die Buchstaben in die Großschreibung umgeschaltet werden. Die Umschaltung von Ziffern auf Sonderzeichen oder von einem auf ein anderes Sonderzeichen wird davon nicht berührt. Der Zustand wird durch eine Leuchtdiode signalisiert.
- BREAK : Durch die Betätigung dieser Taste wird ein Signal erzeugt, das häufig zur Unterbrechung von Übertragungen benutzt wird. Siehe auch Hinweise in Abschnitt 4 dieser Funktionsbeschreibung (break = unterbrechen).

ASCII - Tastatur

4. Anschlußbelegung der Tastatur

Bild 3 zeigt die Anschlußbelegung des 15-poligen Steckers der Tastatur.

Anschluß-Nr.	Bedeutung	Bemerkung
1	+ 5 V	} Spannungsvorsorgung
2	0 V	
3	bit 7	} ASCII - Kodierung
4	bit 6	
5	bit 5	
6	bit 4	
7	bit 3	
8	bit 2	
9	bit 1	
10	ENABLE I	Steuereingang ²⁾
11	STB	Steuerausgang ¹⁾
12	AKD	Steuerausgang ²⁾
13	ENABLE II	Steuereingang ²⁾
14	PARITÄT	Zusatzbit ²⁾
15	BREAK	Steuerausgang ²⁾

1) siehe Erläuterung

2) siehe Hinweise

Bild 3: Anschlußbelegung Tastaturstecker

Erläuterung:

Der Anschluß 11 "STB" bedeutet Strobe-Signal (strobe = Markierung). Wird eine Taste betätigt, so wird nach ca. 2 µs ein Strobe-Impuls von 4 µs Dauer erzeugt. Dieser Impuls dient einem angeschlossenen Empfangsgerät als Signal dafür, daß ein ASCII - Zeichen an den Anschlüssen 3 ... 9 anliegt. Er wird zur Synchronisation zwischen Tastatur und angeschlossenem Gerät benutzt. Wird eine Taste längere Zeit betätigt, so werden nach einer gewissen Verzögerungszeit weitere Strobe-Impulse erzeugt. Das hat zur Folge, daß das Zeichen der betätigten Taste wiederholt wird, solange diese Taste betätigt bleibt.

Hinweise:

Die folgenden zusätzlichen Tastatursignale werden im MC-System nicht ausgenutzt. Auf ihre Funktion wird deshalb nur hingewiesen.

Eingänge:

Anschluß 10, "ENABLE I" : An diesem Signaleingang kann die Ausgabe des Strobe- und des AKD-Signales verhindert werden. (H = Enable; L = disable, Ausgänge werden auf L-Signal gehalten.)

Anschluß 13, "ENABLE II": Durch ein Signal an diesem Eingang können die Datenausgänge gesperrt werden. (H = Enable; L = disable, Datenausgänge werden auf L-Signal gehalten.) enable = ermöglichen; disable = unwirksam machen

Ausgänge:

Anschluß 12, "AKD": Am Ausgang Any Key Down (Taste betätigt) wird ein H-Signal erzeugt, wenn eine Taste betätigt ist. Der Ausgang führt dann wieder L-Signal, wenn keine der Tasten mehr betätigt ist.

Anschluß 14, "PARITÄT": An diesem Ausgang steht das Paritätsbit zur Verfügung. Gebildet wird die "gerade Parität", d.h. das Paritätsbit wird H, wenn die Anzahl der H-Bits in dem 7-Bit-Wort ungerade ist.

Anschluß 15, "BREAK": Der Ausgang "Break" wird direkt durch Betätigung der Break-Taste auf 0 V geschaltet. Diese Funktion wird häufig zur Unterbrechung von Übertragungen benutzt.

Name: _____

ASCII - Tastatur

Datum: _____

Belegungsplan Stecker und Tastatur

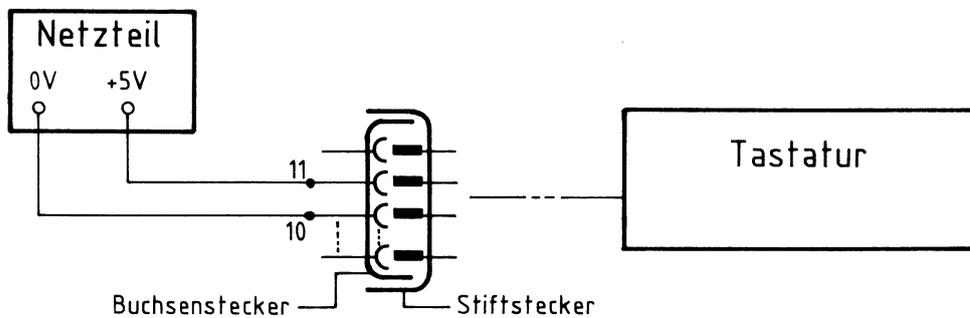
Tastatur-Stift	Ader-farbe	Stecker-Anschlußstift	Summer	Bedeutung
1		11		+ 5 V
2		10		0 V
3		7		bit 7
4		6		bit 6
5		5		bit 5
6		4		bit 4
7		3		bit 3
8		2		bit 2
9		1		bit 1
11		8		STB
		9	rot	Summeransteuerung
		15	schwarz	

Name: _____

ASCII - Tastatur

Datum: _____

Zur Inbetriebnahme der Tastatur ist folgender Meßaufbau vorzubereiten:



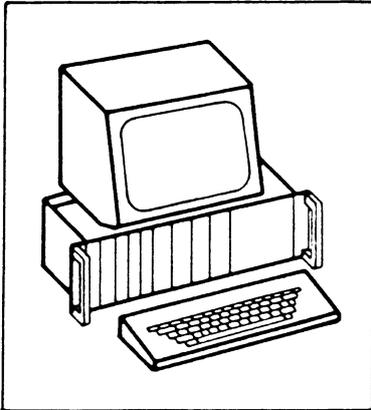
Betätigen Sie nun die LOCK-Taste. Damit haben Sie alle Zeichen auf die obere Funktion umgeschaltet (Großbuchstaben, Satzzeichen etc.). Messen Sie jetzt jeweils nach Anschlag einer der Tasten U,* und L die Pegel der Datenleitungen b1 bis b7. Tragen Sie die Werte in die folgende Tabelle ein. Wenn die von Ihnen gemessenen Pegel den angegebenen Kontrollwerten entsprechen, ist die Tastatur betriebsbereit, andernfalls müssen Sie die Verdrahtung von Tastatur und Stecker überprüfen.

		Meßwerte							Kontrollwerte						
		b7	b6	b5	b4	b3	b2	b1	b7	b6	b5	b4	b3	b2	b1
Datenbit															
Stift- Nr.		7	6	5	4	3	2	1	7	6	5	4	3	2	1
Pegelwerte für:	U								H	L	H	L	H	L	H
	*								L	H	L	H	L	H	L
	L								H	L	L	H	H	L	L

Am Strobe-Anschluß (Stift-Nr. 8) muß mit einem TTL-Tester bei Dauerbetätigung einer Buchstabentaste ein sich periodisch wiederholender Impuls meßbar sein.

Damit ist die Übung beendet.

FACHPRAKTISCHE ÜBUNG MIKROCOMPUTER-TECHNIK



Video-Interface

BFZ/MFA 8.4.

Bildformat:

24 Zeilen / 80 Zeichen





Inhalt

1	Allgemeines	4
2	Funktionsbeschreibung	4
2.1	Einleitung	4
2.2	Darstellung von Zeichen auf dem Bildschirm	5
2.3	Die Erzeugung der Signale für die Zeichendarstellung	8
2.3.1	Einschreiben der Zeichen in den Bildwiederhol-Speicher	8
2.3.2	Auslesen der im Bildwiederhol-Speicher eingeschriebenen Zeichen (Bilddarstellung)	10
2.3.3	Auslesen der Zeichenform-Information aus dem Zeichengenerator	11
2.3.4	Erzeugung des BAS-Signals aus den Punktreihen-Daten	12
2.4	Steuerzeichen und Attribute	12
2.5	Die Gesamtschaltung des Video-Interfaces	13
3	Emulation des Video-Interfaces BFZ/MFA 8.2 und des Terminals TVI 950.....	14
3.1	Der MAT 85-Emulations-Mode	14
3.1.1	Die 1-Byte-Steuerzeichen im MAT 85-Mode	15
	- Summer.....	15
	- Cursor nach links (Backspace).....	15
	- Cursor nach rechts.....	15
	- Cursor nach unten (Line Feed, Zeilenvorschub)	16
	- Cursor nach oben.....	16
	- Bildschirm löschen, Cursor in Home-Position	16
	- Cursor an den Zeilenanfang, Zeilenrest löschen.....	16
	- Zeile löschen	16
	- Cursor nach unten	17
	- Cursor in die Home-Position.....	17
	- Cursor an den Zeilenanfang	17
3.1.2	Die Steuer-Sequenzen im MAT 85-Mode.....	17
	- TVI 950-Mode einschalten	17
	- MAT 85-Mode einschalten	18
	- Aktuellen Mode abfragen	18
	- Software-Reset.....	18
	- Abfrage der Software-Version	18
3.2	Der TVI 950-Emulations-Mode	19
3.2.1	Die 1-Byte-Steuerzeichen im TVI 950-Mode	19
	- Summer.....	19
	- Cursor nach links (Backspace).....	19
	- Tabulator.....	20
	- Cursor nach unten, mit Scrollen (Line Feed, Zeilenvorschub).....	20
	- Cursor nach oben.....	20
	- Cursor nach rechts.....	20
	- Cursor an den Zeilenanfang (Carriage Return, Wagenrücklauf)	21
	- Cursor nach unten, ohne Scrollen	21
	- Bildschirm löschen, Cursor in Home-Position	21
	- Cursor in die Home-Position.....	21

Inhalt

3.2.2	Die Steuer-Sequenzen im TVI 950-Mode.....	21
3.2.2.1	Bildschirm löschen, Cursor Home	22
3.2.2.2	Escape-Sequenzen zum Editieren des Schriftfeldes	22
	- Ein Leerzeichen einfügen	22
	- Ein Zeichen löschen	22
	- Eine Zeile einfügen	22
	- Eine Zeile löschen	23
	- Zeile ab Cursor-Position löschen	23
	- Schriftfeld ab Cursor-Position löschen	23
3.2.2.3	Cursor-Positionierung.....	23
	- Cursor-Position setzen.....	23
	- Cursor-Position abfragen	24
	- Cursor nach links zum nächsten Tabulator (Backtab)	25
3.2.2.4	Zeichensätze, Zeichendarstellung (Attribute), Cursor-Darstellung	25
	- Zeichensatz wählen.....	25
	- Attribute setzen/löschen	27
	- Halbe Helligkeit.....	29
	- Volle Helligkeit.....	29
	- Bildhintergrund hell	29
	- Bildhintergrund dunkel.....	29
	- Control-Mode einschalten.....	29
	- Cursor-Darstellung	30
3.2.2.5	Die DLE-Sequenzen im TVI 950-Mode	31
	- TVI 950-Mode einschalten	31
	- MAT 85-Mode einschalten.....	31
	- Aktuellen Mode abfragen	31
	- Software-Reset.....	31
	- Abfrage der Software-Version	32
4	Einsatz unter dem Betriebsprogramm MAT 85.....	33
4.1	Schalterstellungen.....	34
4.2	Nutzung in Assembler-Programmen	35
4.2.1	Zeichenausgabe	35
4.2.2	Zeicheneingabe.....	36
4.2.3	Ein Programmbeispiel in Assembler-Sprache.....	37
4.3	Nutzung unter BFZ-STEUER-BASIC	39
4.3.1	Zeichenausgabe	39
4.3.2	Zeicheneingabe.....	40
4.3.3	Ein Programmbeispiel in BFZ-STEUER-BASIC	41

Inhalt

5	Einsatz unter dem Betriebssystem CP/M.....	43
5.1	Einstellung der Schalter des Video-Interfaces.....	46
5.2	Nutzung in Assembler-Programmen	48
5.2.1	Zeichenausgabe	48
5.2.2	Zeicheneingabe.....	49
5.2.3	Ein Programmbeispiel in Assembler-Sprache.....	52
5.3	Nutzung unter MBASIC (BASIC 80).....	55
5.3.1	Zeichenausgabe	55
5.3.2	Zeicheneingabe.....	56
5.3.3	Ein Programmbeispiel in MBASIC (BASIC 80).....	57
5.4	Das Editor-Programm BFZED.COM.....	58
5.4.1	Vorbereitungen zum Anpassen des Programms BFZED.COM.....	59
5.4.2	Anpassen des Programms BFZED.COM	59
6	Die Statuszeile	62
7	Das Set-Up-Menü.....	64
8	Der Control-Mode	67

Anhang

A1	Stromlaufplan	68
A2	Bestückungsplan.....	70
A3	Bauteilliste.....	71
A4	Lötbrückentabelle	72
A5	Steckerbelegungen.....	74
A6	ASCII-Tabelle	75
A7	Tabelle der länderspezifischen Zeichen	76
A8	Darstellung der Steuerzeichen im Control-Mode	77
A9	Bedeutung der Schalterstellungen der Schalter-Reihen S1 bis S3.....	78
A10	Tabelle der Attribut-Bytes für die Escape-Sequenz "ESC G".....	82
A11	Steuerzeichen im MAT 85-Mode	84
	- 1-Byte-Steuerzeichen	84
	- DLE-Sequenzen	84
A12	Steuerzeichen im TVI 950-Mode.....	85
	- 1-Byte-Steuerzeichen	85
	- DLE-Sequenzen	85
	- Escape-Sequenzen (nach Funktionen geordnet).....	86
	- Escape-Sequenzen (nach Codes geordnet).....	87
A13	Stichwortverzeichnis	88
A14	Verzeichnis der Abbildungen	91

CP/M ist ein eingetragenes Warenzeichen der Firma **Digital Research**

MBASIC ist ein eingetragenes Warenzeichen der Firma **Microsoft**

WORDSTAR ist ein eingetragenes Warenzeichen der Firma **MICROPRO**

TVI 950 ist ein eingetragenes Warenzeichen der Firma **TeleVideo Systems**

Funktionsbeschreibung

1 Allgemeines

Das hier beschriebene Video-Interface BFZ/MFA 8.4 ersetzt das Video-Interface BFZ/MFA 8.2. Es bietet alle Funktionen des Interfaces BFZ/MFA 8.2 und kann darüber hinaus Kleinbuchstaben und Umlaute darstellen. Auf dem Video-Interface ist ein Summer integriert. Er übernimmt die Funktion des in der Tastatur eingebauten Summers. In der zusätzlichen Betriebsart "TVI 950" sind Steuersequenzen implementiert, die z.B. die Positionierung des Cursors und die Wahl verschiedener Darstellungsarten ermöglichen.

Das Video-Interface kann unter dem Betriebsprogramm MAT 85 und unter dem Betriebssystem CP/M verwendet werden. An dem Betriebsprogramm/Betriebssystem des MFA-Mikrocomputers sind keine Änderungen erforderlich. Eine Hardwareänderung (Drahtbrücken auf der verwendeten Serien-Schnittstelle) ist lediglich beim Einsatz unter dem Betriebssystem CP/M erforderlich. Grundvoraussetzung für ein ordnungsgemäßes Funktionieren (gleichgültig ob unter MAT 85 oder unter CP/M) ist allerdings die richtige Einstellung der DIL-Schalter auf dem Video-Interface BFZ/MFA 8.4. Diese Einstellung hängt vom verwendeten Betriebsprogramm ab und ist im Kapitel "Einsatz unter dem Betriebsprogramm MAT 85" bzw. im Kapitel "Einsatz unter dem Betriebssystem CP/M" beschrieben.

2 Funktionsbeschreibung

2.1 Einleitung

Für den Austausch von Informationen zwischen Computer und Bediener verwendet man *Datensichtstationen*. In Computersystemen kommen Datensichtstationen meist dann zur Anwendung, wenn Textinformationen angezeigt oder eingegeben werden müssen.

Eine Datensichtstation besteht aus den Funktionseinheiten *Tastatur*, *Video-Interface* und *Monitor*. Der Monitor entspricht einem Fernsehgerät ohne HF-Empfangsteil, da das im Video-Interface erzeugte Bildsignal (auch BAS-Signal genannt) unmittelbar dem Monitor zugeführt wird. Bild 1 zeigt die Zusammenschaltung einer Datensichtstation mit einem Mikrocomputer.

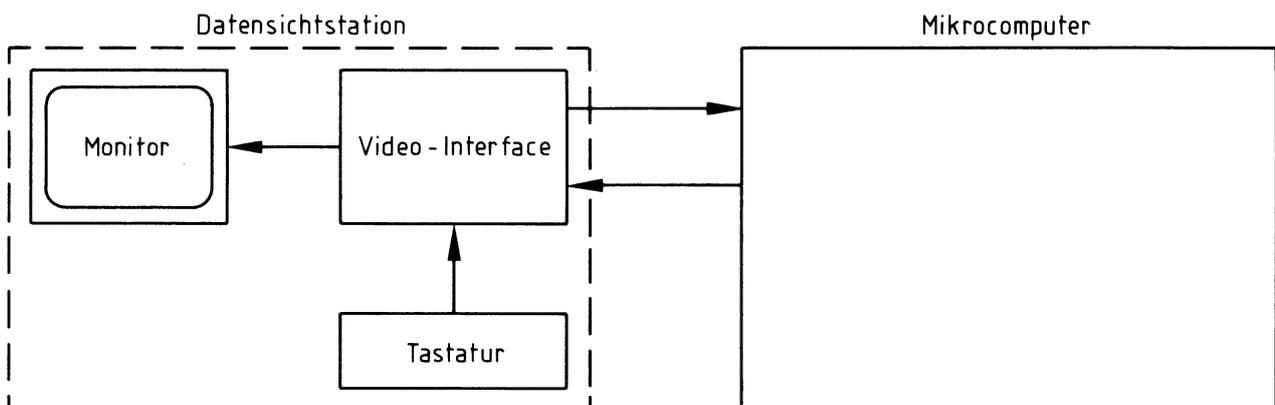


Bild 1: Datensichtstation und Mikrocomputer

Funktionsbeschreibung

Jede Betätigung einer Taste der Tastatur bewirkt, daß ein Zeichen von der Datensichtstation an den Mikrocomputer gesendet wird. Andererseits werden Zeichen, die vom Mikrocomputer an die Datensichtstation gesendet werden, auf dem Bildschirm dargestellt. Für diesen Betrieb ist im Mikrocomputer ein Programm erforderlich, das Teil des Betriebssystems bzw. der Anwender-Software ist.

2.2 Darstellung von Zeichen auf dem Bildschirm

Für die Darstellung von Zeichen (Buchstaben, Ziffern, Sonderzeichen) auf dem Bildschirm wird ein Verfahren angewendet, bei dem jedes Zeichen durch Punkte eines *Matrixrasters* dargestellt wird. Häufig werden hierzu Matrixgrößen von 5x9 oder 7x10 Punkten verwendet. Bild 2 zeigt ein 5x9-Raster für die Buchstaben H und y.

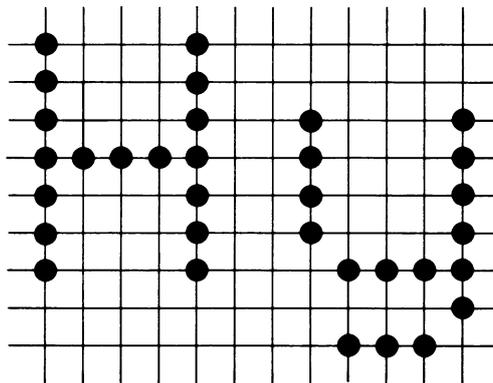


Bild 2: Zeichendarstellung im 5 × 9-Raster

Die einzelnen Punkte - auch *Pixel* genannt (vom engl. Picture Element, Bild-Element) - werden im Monitor durch einen Elektronenstrahl erzeugt, der entsprechend den darzustellenden Zeichenpunkten in seiner Intensität gesteuert wird (Hell-/Dunkelsteuerung). Dieser Strahl wandert kontinuierlich Punktreihe für Punktreihe über die Leuchtschicht der Bildröhre.

Zur Strahlablenkung enthält der Monitor zwei Sägezahngeneratoren. Die horizontale Ablenkung übernimmt ein Sägezahngenerator mit der Periodendauer $T_1 = 64 \mu\text{s}$. Die vertikale Ablenkung erfolgt durch einen zweiten Sägezahngenerator mit der Periodendauer $T_2 = 20 \text{ ms}$. Aus dem Verhältnis der beiden Periodenzeiten T_2/T_1 ergibt sich die Anzahl von 312,5 Punktzeilen je Bild. Innerhalb einer Sekunde wird das gesamte Schriftfeld 50 mal auf dem Bildschirm dargestellt. Die Sägezahngeneratoren im Monitor werden durch Horizontal- und Vertikal-Impulse synchronisiert. Bild 3 zeigt Lage und Einteilung des Schriftfeldes auf dem Bildschirm.

Funktionsbeschreibung

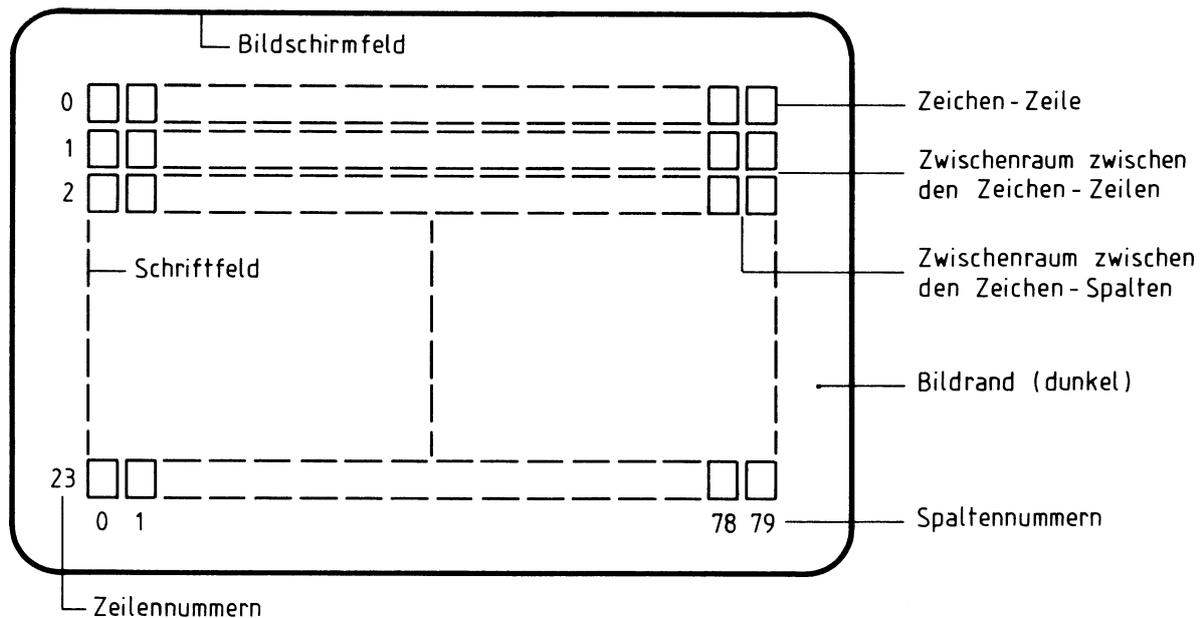


Bild 3: Schriftfeld auf dem Bildschirm

Das hier beschriebene Video-Interface kann wahlweise 22, 24, 26 oder 28 Zeichen-Zeilen und 72, 80, 88 oder 96 Zeichen-Spalten auf dem Bildschirm darstellen. Im folgenden wird jeweils von einer Einstellung auf 24 Zeichen-Zeilen und 80 Zeichen-Spalten ausgegangen. Diese Einstellung wurde ebenfalls für das Bild 3 angenommen. Zwischenräume zwischen den Zeichen sorgen für eine gute Lesbarkeit. Zur Vermeidung von Randunschärfen ist das Schriftfeld kleiner als das Bildschirmfeld. Dazu wird der Elektronenstrahl im Bereich der Bildschirmränder dunkelgesteuert.

Damit die einzelnen Zeichen - und damit das gesamte Schriftfeld - unverzerrt und ruhig auf dem Bildschirm erscheinen, muß das Video-Interface folgende Informationen an den Monitor liefern:

- Impulse, die dem Monitor anzeigen, daß ein neues Bild beginnen soll. Durch diese Impulse wird der Elektronenstrahl dunkelgesteuert und zur linken oberen Bildecke abgelenkt (Vertikal-Synchronimpulse).
- Impulse, die dem Monitor anzeigen, daß eine neue Punktzeile beginnen soll. Hierdurch wird der Elektronenstrahl dunkelgesteuert und zum nächsten Zeilenbeginn abgelenkt (Horizontal-Synchronimpulse).
- Die Information über die hell oder dunkel darzustellenden Bildpunkte einer Zeile (Bild-Signal).

Diese Informationen müssen entsprechend der CCIR-Fernsehnorm in einem Signal vermischt dem Monitor zugeführt werden. Dieses Signal nennt man kurz BAS-Signal, entsprechend der in ihm enthaltenen Informationen für Bild, Austastung (Dunkelsteuerung) und Synchronisierung.

Funktionsbeschreibung

Bild 4 zeigt den Verlauf des Signals, das zur Darstellung der Punkte einer Punktreihe benötigt wird. 11 solcher "Signalpakete" werden zur Darstellung einer Zeichen-Zeile benötigt (9 zur Darstellung der jeweils 9 Pixel hohen Zeichen, 2 als Abstand zur nächsten Zeichen-Zeile). 264 "Signalpakete" werden zur Darstellung des gesamten Schriftfeldes vom Video-Interface erzeugt (24 Zeichen-Zeilen mit je 11 Punktzeilen).

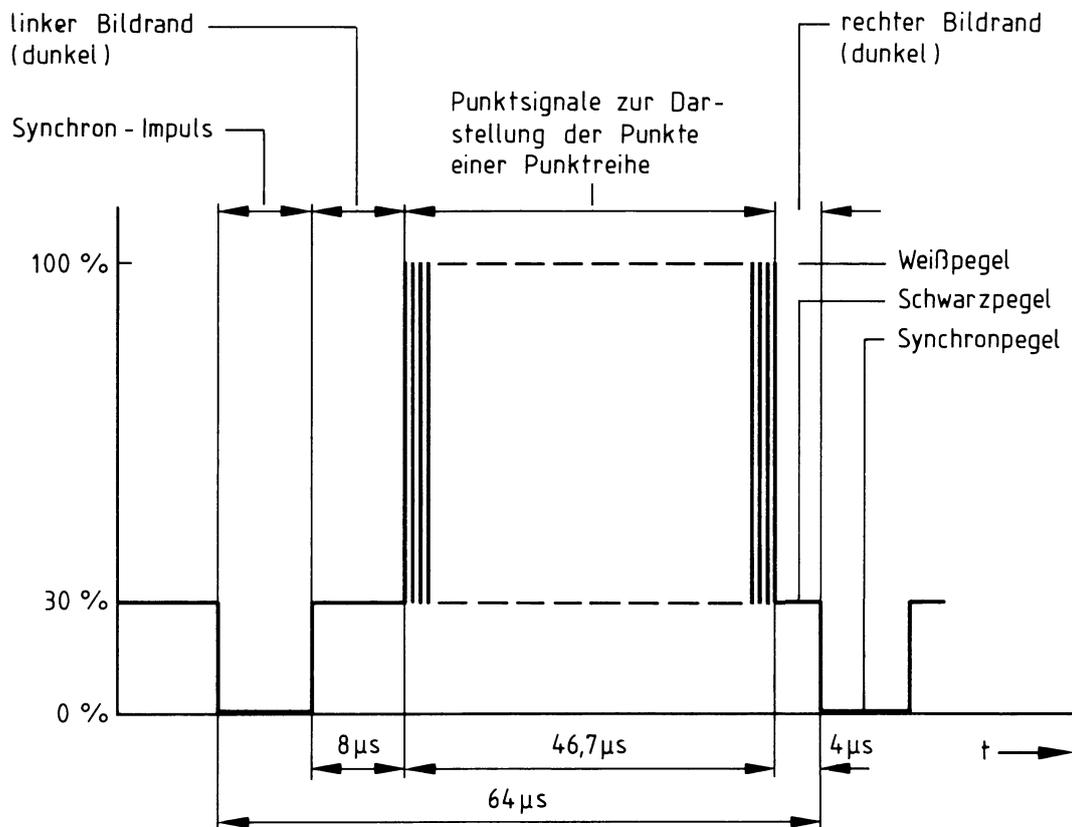


Bild 4: Das BAS - Signal

In einer Zeichen-Zeile werden 80 Zeichen (jeweils 5 Punkte breit) dargestellt. Zusammen mit den zwei Punkten zwischen den Zeichen ergeben sich so 560 Punkte je Punkt-Zeile. Zur Darstellung dieser Punkte innerhalb von $46,7 \mu\text{s}$ benötigt man einen "Punktgenerator" mit einer Frequenz von 12 MHz.

Funktionsbeschreibung

2.3 Die Erzeugung der Signale für die Zeichendarstellung

Das Blockschaltbild (Bild 5) zeigt die Haupt-Funktionseinheiten der Schaltung.

- Der CRT-Controller
Er steuert die Zeichenausgabe und erzeugt die hierfür notwendigen Signale. (CRT = cathode ray tube, Bildröhre. Controller = Steuerbaustein).
- Der Bildwiederhol-Speicher
Der Bildwiederhol-Speicher besteht aus dem *Zeichencode-Speicher* und dem *Attribut-Speicher*. Im Zeichencode-Speicher werden die Codes der auf dem Bildschirm darzustellenden Zeichen gespeichert. Durch die Speicher-Adresse innerhalb des Zeichencode-Speichers ist die Position des Zeichens auf dem Bildschirm festgelegt. Im Attribut-Speicher werden die *Attribute* der einzelnen Zeichen gespeichert. Die Attribute legen fest, wie ein Zeichen darzustellen ist (blinkend, unterstrichen usw.). Die über Attribute festgelegte *Darstellungsart* darf nicht mit der durch den Zeichengenerator festgelegten *Zeichenform* verwechselt werden. Der Zeichencode- und der Attribut-Speicher haben gemeinsame Adreß- aber getrennte Daten-Leitungen. Am Zeichencode- und am Attribut-Speicher liegt also immer die gleiche Adresse an.
- Der Zeichengenerator (Character-Generator)
Der Zeichengenerator ist ein Nur-Lese-Speicher (ROM). Durch ihn wird die Form der einzelnen Zeichen festgelegt.
- Der Mikrocontroller
Der Mikrocontroller ist ein in einem IC zusammengefaßter Mikrocomputer (auch Single-Chip-Mikrocomputer genannt). Er wickelt den Datenaustausch zwischen Interface und MFA-Mikrocomputer ab. Der Mikrocontroller erkennt die vom Mikrocomputer zum Interface gesendeten Steuer-Zeichen und führt die gewünschten Funktionen (z.B. Bildschirm löschen) aus. Werden darstellbare Zeichen zum Interface übertragen, so "weiß" der Mikrocontroller, an welcher Stelle des Schriftfeldes die empfangenen Zeichen dargestellt werden sollen und schreibt sie in die entsprechende Speicher-Zeile des Bildwiederhol-Speichers ein.

2.3.1 Einschreiben der Zeichen in den Bildwiederhol-Speicher

Die Zeichen, die auf dem Bildschirm dargestellt werden sollen, sind im ASCII-Code verschlüsselt (ASCII = American Standard Code for Information Interchange, amerikanischer Standard-Code für den Informationsaustausch). Eine ASCII-Codetabelle befindet sich im Anhang.

Das erste Zeichen, das der Mikrocomputer nach dem Einschalten zum Interface überträgt, wird in der linken oberen Ecke des Schriftfeldes dargestellt. Soll dort z.B. der Buchstabe "A" dargestellt werden, so sendet der Mikrocomputer den Code 41_{H} an das Video-Interface. Der Mikrocontroller auf dem Interface erkennt aus dem Code, daß es sich um ein darstellbares Zeichen handelt. Er "weiß" außerdem, an welcher Position des Schriftfeldes das Zeichen dargestellt werden soll und

Funktionsbeschreibung

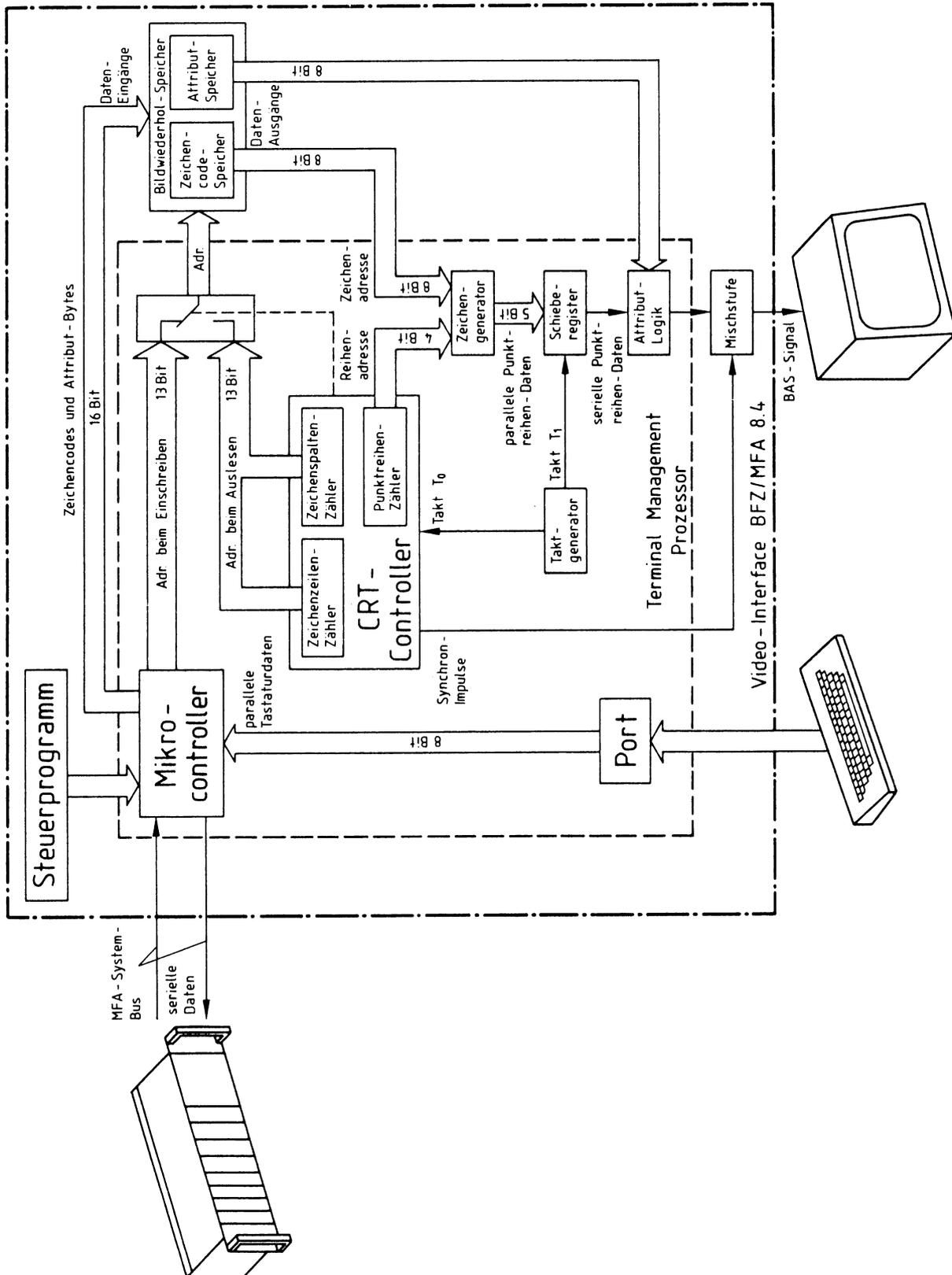


Bild 5: Blockschaltbild des Video-Interfaces

Funktionsbeschreibung

trägt den Code in die entsprechende Speicherzeile (in diesem Falle in die erste Speicher-Zeile) des Zeichencode-Speichers ein. Überträgt der Mikrocomputer als nächstes Zeichen den Buchstaben "B" (Code 42_{H}) an das Video-Interface, so wird dieser Code in die nächste Speicher-Zeile des Zeichencode-Speichers geschrieben.

Hat der Mikrocomputer nach dem Einschalten zwei Zeichen (z.B. "A" und "B") zum Interface übertragen, so werden diese beiden Zeichen nebeneinander in der ersten Zeile des Schriftfeldes angezeigt. Das dritte Zeichen, das der Mikrocomputer zum Interface überträgt, würde rechts neben dem "B" - also in der dritten Spalte der **ersten** Zeile - angezeigt. Will man nun erreichen, daß dieses Zeichen in der dritten Spalte der **zweiten** Zeile angezeigt wird, so muß man zuvor das Steuerzeichen "Line Feed" (Zeilenvorschub, Code $0A_{\text{H}}$) zum Interface übertragen. Der Mikrocontroller erkennt dieses Steuerzeichen und trägt das darauf folgende Zeichen in die richtige Speicher-Zeile des Zeichencode-Speichers ein. Auf dem Bildschirm kennzeichnet ein *Cursor* (Schreibmarke) die Stelle, an der das nächste Zeichen angezeigt wird.

2.3.2 Auslesen der im Bildwiederhol-Speicher eingeschriebenen Zeichen (Bilddarstellung)

Während der Elektronenstrahl eine Punktreihe überstreicht, muß er entsprechend den darzustellenden Zeichen hell- oder dunkelgesteuert werden. Der dazu notwendige Ablauf ist im folgenden beschrieben:

Der CRT-Controller besitzt intern mehrere Zähler. Einer davon zählt, welche Zeichen-Spalte momentan vom Elektronenstrahl auf dem Bildschirm geschrieben wird (Zeichenspalten-Zähler). Ein weiterer zählt, welche Zeichen-Zeile momentan vom Elektronenstrahl geschrieben wird (Zeichenzeilen-Zähler). Um einen Zeichen-Code aus dem Zeichencode-Speicher auszulesen, schaltet der CRT-Controller **beide** Zählerstände auf die Adressleitungen des Speichers (siehe Bild 5). Auf diese Weise wird im Zeichencode-Speicher das Zeichen adressiert, das momentan vom Elektronenstrahl auf den Bildschirm angezeigt wird.

Wird zum Beispiel gerade das erste Zeichen der ersten Zeile dargestellt, so stehen beide Zähler auf Null, da die Zeilen- und Spalten-Zählung jeweils bei Null beginnt. Daraus ergibt sich die Speicher-Adresse 0000_{H} . Soll an dieser Bildschirmposition der Buchstabe "A" dargestellt werden, so ist zuvor vom Mikrocontroller der entsprechende ASCII-Code (41_{H}) in diese Speicher-Zeile des Zeichencode-Speichers eingeschrieben worden. Der Code wird nun vom CRT-Controller ausgelesen und liegt an den Daten-Ausgängen des Zeichencode-Speichers an. Durch diesen Code ist lediglich festgelegt, welches Zeichen dargestellt werden soll. Wie dieses Zeichen dargestellt werden soll, also die Zeichenform, wird durch den nachfolgend beschriebenen Zeichengenerator festgelegt.

Funktionsbeschreibung

2.3.3 Auslesen der Zeichenform-Information aus dem Zeichengenerator

Die Information über die Form des darzustellenden Zeichens ist im Zeichengenerator festgelegt. Hierbei handelt es sich um einen Nur-Lese-Speicher (ROM). Bild 6 zeigt den Inhalt einiger Speicher-Zeilen dieses ROMs. Die dargestellten Speicher-Zeilen enthalten die Information über die Form der Buchstaben "G" und "g". Die aus dem Zeichencode-Speicher ausgelesenen Daten (ASCII-Code der darzustellenden Zeichen) gelangen auf die höherwertigen Adreßleitungen des Zeichengenerators. Über diese Adreßleitungen wird die Zeichenform-Information für das darzustellende Zeichen ausgewählt. Man spricht daher auch von der *Zeichen-Adresse*.

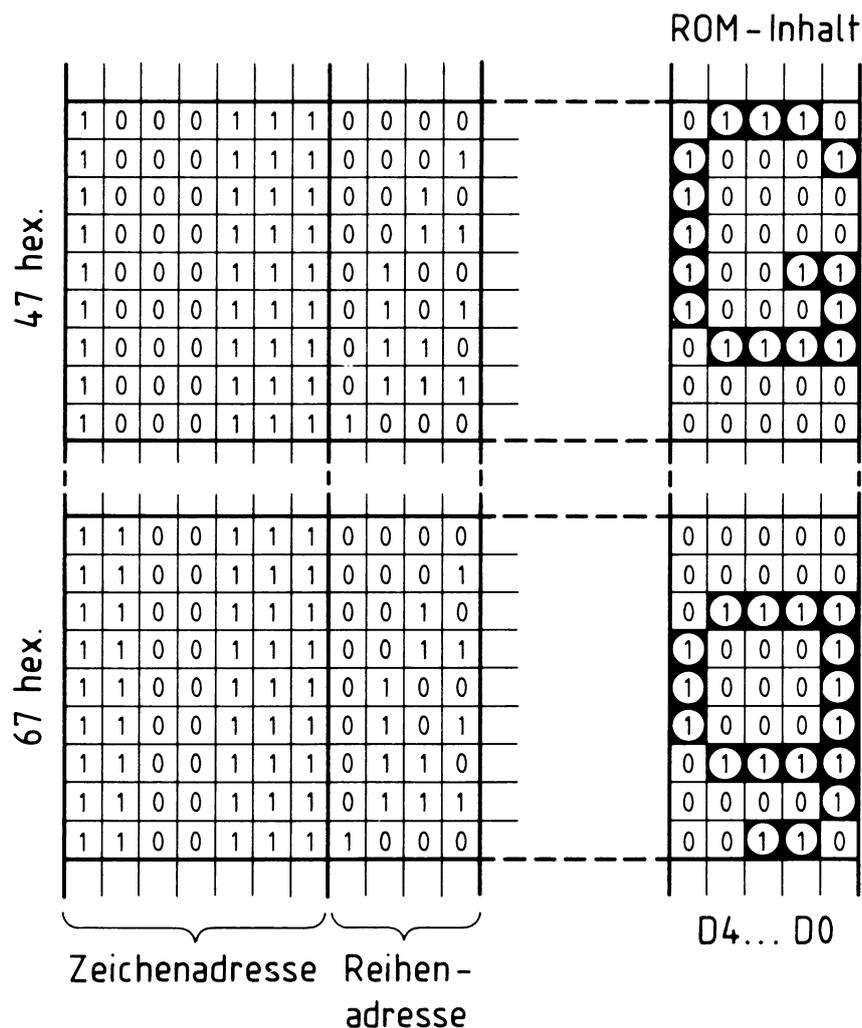


Bild 6: Programmierung des Zeichengenerators (Auszug)

Die einzelnen Zeichen sind aus 9 Punktreihen aufgebaut. Während der Elektronenstrahl von links nach rechts über den Bildschirm abgelenkt wird, zeichnet er jeweils eine dieser Punktreihen. Ein weiterer Zähler innerhalb des CRT-Controllers enthält hierzu die Information, in welcher Punktreihe des Zeichens der Elektronenstrahl sich momentan befindet. Der Stand dieses Punktreihen-Zählers gelangt auf die niederwertigen Adressleitungen des Zeichengenerators (Reihen-Adresse). Über die Reihenadresse wird die Zeichenform-Information einer Punktreihe (Punkt-

Funktionsbeschreibung

reihen-Daten) des darzustellenden Zeichens ausgewählt. Die Daten einer Punktreihe werden aus dem Zeichengenerator ausgelesen und gelangen zum Schieberegister.

2.3.4 Erzeugung des BAS-Signals aus den Punktreihen-Daten

Zur Ansteuerung des Monitors muß aus den Punktreihen-Daten noch das BAS-Signal gewonnen werden. Hierzu setzt ein Schieberegister die Punktreihen-Daten in einen seriellen Datenstrom um. Während der Elektronenstrahl die Zeichen-Spalte überstreicht, werden nacheinander alle 5 Punkte der Punktreihe des darzustellenden Zeichens am Ausgang des Schieberegisters ausgegeben.

Das Schieberegister liefert aber nur den Bild-Anteil des BAS-Signals. Die Austast- und Synchron-Impulse werden vom CRT-Controller erzeugt. Die Mischstufe vereinigt alle Signal-Anteile zum BAS-Signal (siehe auch Bilder 4 und 5).

2.4 Steuerzeichen und Attribute

- Steuerzeichen

Um das Video-Interface vielseitig nutzen zu können, überträgt der Mikrocomputer außer den darzustellenden Zeichen auch sogenannte Steuerzeichen zum Interface. Eines dieser Steuerzeichen ist "Carriage Return" mit dem Code $0D_H$. Nach der Übertragung dieses Steuerzeichens wird das nächste darstellbare Zeichen am linken Rand des Schriftfeldes angezeigt. Die Steuerzeichen werden nicht in den Bildwiederhol-Speicher eingetragen. Der Mikrocontroller des Video-Interfaces filtert diese Zeichen aus dem empfangenen Datenstrom heraus und führt die entsprechenden Funktionen aus.

- Attribute

Das Video-Interface bietet vielfältige Darstellungsarten der Zeichen: invers, unterstrichen, blinkend, doppelt hoch, doppelt breit, halbe Helligkeit. Diese Darstellungsarten werden durch sogenannte *Attribute* festgelegt. Es ist möglich, einem Zeichen mehrere Attribute - z.B.: unterstrichen und blinkend - zuzuordnen. Die Attribute werden vom Mikrocontroller in einen besonderen Teil des Bildwiederhol-Speichers, dem Attribut-Speicher, eingetragen. In diesem Speicher ist jeder Zeichencode-Speicherzeile eine Attribut-Speicherzeile zugeordnet. In jedem Bit der Attribut-Speicherzeile ist die Ein-/Aus-Information für eines der möglichen Darstellungsarten festgehalten. Theoretisch können einem Zeichen so acht verschiedene Attribute zugeordnet werden - die Steuersoftware des Mikrocontrollers unterstützt jedoch nur sieben.

Die Attribute werden über Steuerzeichen-Folgen ein- bzw. ausgeschaltet. Die eingeschalteten Attribute gelten für alle Zeichen, die bis zum Ausschalten der entsprechenden Attribute übertragen werden.

Funktionsbeschreibung

Beispiel:

1. Steuerzeichen-Folge: NUR "UNTERSTREICHEN" EINSCHALTEN
Alle folgenden Zeichen werden unterstrichen dargestellt
2. Steuerzeichen-Folge: ZUSÄTZLICH "BLINKEN" EINSCHALTEN
Alle folgenden Zeichen werden unterstrichen und blinkend dargestellt
3. Steuerzeichen-Folge: "UNTERSTREICHEN" AUSCHALTEN (andere Attribute unverändert)
Alle folgenden Zeichen blinken, werden aber nicht unterstrichen dargestellt
4. Steuerzeichen-Folge: "BLINKEN" AUSCHALTEN
Alle folgenden Zeichen werden nun normal (nicht blinkend) dargestellt

Der Mikrocontroller erkennt die Steuerzeichen zum Ein- bzw. Ausschalten der Attribute und merkt sich die aktuellen Zustände (ein/aus) der Attribute. Immer wenn er einen Zeichen-Code in den Zeichencode-Speicher einträgt, schreibt er parallel dazu die Zustände der Attribute in den Attribut-Speicher.

Soll ein im Bildwiederhol-Speicher befindliches Zeichen auf dem Bildschirm dargestellt werden, so wird der Zeichen-Code aus dem Zeichencode-Speicher ausgelesen. Parallel dazu wird das zugehörige Attribut-Byte aus dem Attribut-Speicher ausgelesen. Das Attribut-Byte steuert die *Attribut-Logik*. Diese Logik befindet sich im Signalweg zwischen Schieberegister und Mischstufe (siehe auch Bild 5). Bei Zeichen mit dem Attribut "INVERS" sorgt diese Logik z.B. dafür, daß die einzelnen Punktreihen-Daten des Zeichens invertiert werden.

2.5 Die Gesamtschaltung des Video-Interfaces

Der Stromlaufplan im Anhang zeigt die Gesamtschaltung des Video-Interfaces.

Der Mikrocontroller, sein interner RAM-Speicher (64x8 Bit), der CRT-Controller, der Zeichengenerator, die Attributlogik, das Schieberegister und der Taktgenerator befinden sich zusammen auf einem Chip - dem Terminal-Management-Processor (TMP) NS 405 A12N. Zusätzlich zu den erwähnten Funktionsgruppen befinden sich noch ein paralleler 8-Bit-I/O-Port und ein Baudraten-generator im TMP.

Der Bildwiederhol-Speicher besteht aus zwei RAM-Bausteinen des Typs 6264 (D2 und D3, Speicherkapazität: je 8-K-Byte) für Zeichencode- und Attribut-Speicher. Als Programm-Speicher für den im TMP integrierten Mikrocontroller dient ein 8-K-Byte-EPROM des Typs 2764 (D11).

Die Mischstufe wird aus dem Antivalenz-Gatter D16, dem Transistor V1, der Diode V3 und einigen Widerständen gebildet.

Da die Übertragung der Daten zwischen Mikrocomputer und Interface seriell mit V.24-Pegel erfolgt, besitzt das Interface noch zwei Pegelwandler zur Wandlung von TTL- in V.24-Pegel und umgekehrt (V2 und D14). Außerdem befinden sich drei Schalterreihen auf der Baugruppe. Über diese Schalter können das Übertragungsformat zwischen Interface und Mikrocomputer sowie die verschiedenen Betriebsarten des Interfaces gewählt werden.

3 Emulation des Video-Interfaces BFZ/MFA 8.2 und des Terminals TVI 950

Auf dem Markt gibt es verschiedene Datensichtstationen. Diese arbeiten teilweise mit unterschiedlichen Steuerzeichen. Das hier beschriebene Video-Interface kann in zwei verschiedenen Modi arbeiten. In jedem dieser Modi wird eine andere Datensichtstation emuliert (nachgebildet). Im ersten Mode (dem MAT 85-Mode) wird das Video-Interface BFZ/MFA 8.2 mit seinen Steuerzeichen, im zweiten Mode (dem TVI 950-Mode) wird das Terminal TVI 950 der Firma TeleVideo emuliert.

In beiden Fällen handelt es sich um eine Teil-Emulation. So werden z.B. im MAT 85-Mode immer mehr Zeichen-Zeilen und Zeichen-Spalten als im emulierten Video-Interface angezeigt (die Anzahl der Zeilen und Spalten ist einstellbar). Im TVI 950-Mode werden nur die wichtigsten Steuerzeichen dieses Terminals unterstützt.

Über den Schalter S1.1 der Interface-Baugruppe kann eingestellt werden, welcher Mode nach dem Einschalten aktiv sein soll. Der Mode kann während des Betriebs beliebig häufig über die Tastatur oder durch Steuerzeichen gewechselt werden.

In den folgenden Kapiteln wird zwischen darstellbaren Zeichen und Steuerzeichen unterschieden. Überträgt man den Code eines darstellbaren Zeichens vom Mikrocomputer zum Interface, wird das entsprechende Zeichen auf dem Bildschirm angezeigt. Hierbei ist die Art der Zeichendarstellung vom eingestellten *Zeichensatz* abhängig. Die Wahl des Zeichensatzes legt z.B. fest, ob deutsche oder französische Sonderzeichen angezeigt werden sollen. Überträgt man den Code eines Steuerzeichens zum Interface, wird die entsprechende Steuerfunktion ausgeführt.

In der Beschreibung der einzelnen Steuerfunktionen wird häufig der Begriff "*Cursor*" verwendet. Hierbei handelt es sich um eine Schreibmarke, die dem Anwender anzeigt, an welcher Stelle des Schriftfeldes das nächste zum Interface übertragene Zeichen dargestellt wird. Die Form dieser Schreibmarke (Strich oder Block) kann eingestellt werden.

3.1 Der MAT 85-Emulations-Mode

Dieser Mode ist für den Betrieb unter dem Betriebsprogramm MAT 85 des MFA-Mikrocomputers vorgesehen. In diesem Mode können alle Steuerzeichen des Video-Interfaces BFZ/MFA 8.2 benutzt werden. Bei vier dieser Steuerzeichen reagiert das hier beschriebene Interface (BFZ/MFA 8.4) geringfügig anders. Dies führt jedoch i. allg. zu keinerlei Problemen.

Zusätzlich zu den oben erwähnten Steuerzeichen, "kennt" das hier beschriebene Interface weitere Steuerzeichen-Folgen (Sequenzen). Mit diesen Sequenzen kann man z.B. von einem Mode in den anderen schalten oder einen Software-Reset (Kaltstart) auslösen.

Unter dem Betriebssystem CP/M sollte der TVI 950-Mode dem MAT 85-Mode vorgezogen werden, da dieser mehr Möglichkeiten (z.B. direkte Cursor-Positionierung) bietet.

 MAT 85-Mode

3.1.1 Die 1-Byte-Steuerzeichen im MAT 85-Mode

Die Steuerfunktionen des Interfaces werden im MAT 85-Mode sowohl durch einzelne Steuerzeichen als auch durch Steuerzeichen-Folgen ausgelöst. In diesem Kapitel sind die einzelnen Steuerzeichen (Control-Codes) beschrieben. Eine Beschreibung der Steuerzeichen-Folgen (Sequenzen) finden Sie im nächsten Kapitel.

Um die entsprechende Steuerfunktion auszulösen, muß der Code des Steuerzeichens vom Mikrocomputer zum Video-Interface übertragen werden. Näheres dazu entnehmen Sie bitte dem Kapitel "Einsatz unter dem Betriebsprogramm MAT 85".

Für jedes Steuerzeichen ist im folgenden seine Bezeichnung, sein hexadezimaler Code, sein dezimaler Code und seine Funktion angegeben.

Summer

BEL	07_H	7_D
------------	-----------------------	----------------------

Durch die Ausgabe des Steuerzeichens BEL ertönt der im Interface eingebaute Summer für etwa 0,5 Sekunden.

Cursor nach links (Backspace)

BS	08_H	8_D
-----------	-----------------------	----------------------

Der Cursor wird um eine Spalte nach links bewegt. Befindet er sich am linken Schriftfeld-Rand, so bewegt er sich an das Ende der vorstehenden Zeile. Wenn der Cursor sich in der linken oberen Ecke des Schriftfeldes befindet, hat dieses Steuerzeichen keine Wirkung.

Dieses Steuerzeichen hat bei dem hier beschriebenen Interface eine geringfügig andere Wirkung als beim Video-Interface BFZ/MFA 8.2: Befindet sich der Cursor beim Interface BFZ/MFA 8.2 links oben im Schriftfeld, so bewegt er sich durch das Steuerzeichen BS zur rechten unteren Ecke.

Cursor nach rechts

HT	09_H	9_D
-----------	-----------------------	----------------------

Der Cursor wird um eine Spalte nach rechts bewegt. Steht der Cursor am Ende einer Zeile, springt er an den Anfang der nächsten Zeile. Wenn der Cursor rechts unten steht, wird das Schriftfeld um eine Zeile nach oben geschoben (gescrollt). Der Inhalt der obersten Zeichen-Zeile geht verloren, unten wird eine Leerzeile eingefügt. Der Cursor springt an den Anfang dieser Leerzeile.

Dieses Steuerzeichen hat bei dem hier beschriebenen Interface eine geringfügig andere Wirkung als beim Video-Interface BFZ/MFA 8.2: Beim Interface 8.2 wird die oben herausgeschobene Zeichen-Zeile anstelle der Leerzeile unten eingefügt.

MAT 85-Mode

Cursor nach unten (Line Feed, Zeilenvorschub)

LF **0A_H** **10_D**
Der Cursor wird - unter Beibehaltung der Spaltenposition - um eine Zeile nach unten bewegt. Steht der Cursor am unteren Schriftfeld-Rand, wird das Schriftfeld um eine Zeile nach oben geschoben. Dabei geht der Inhalt der obersten Zeichen-Zeile verloren. Unten wird eine Leerzeile eingefügt.

Cursor nach oben

VT **0B_H** **11_D**
Der Cursor wird - unter Beibehaltung der Spaltenposition - um eine Zeile nach oben bewegt. Steht der Cursor am oberen Schriftfeld-Rand, wird dieses Steuerzeichen ignoriert.

Dieses Steuerzeichen hat bei dem hier beschriebenen Interface eine geringfügig andere Wirkung als beim Video-Interface BFZ/MFA 8.2: Befindet sich der Cursor beim Interface 8.2 am oberen Schriftfeld-Rand, springt er - unter Beibehaltung der Spaltenposition - an den unteren Schriftfeldrand.

Bildschirm löschen, Cursor in Home-Position

FF **0C_H** **12_D**
Der Bildschirm wird mit Leerzeichen gefüllt (gelöscht). Der Cursor wird in die obere linke Ecke des Schriftfeldes bewegt (Home-Position).

Cursor an den Zeilenanfang, Zeilenrest löschen

CR **0D_H** **13_D**
Der Cursor wird an den Zeilenanfang bewegt. Die Zeichen ab der alten Cursor-Position (einschließlich) bis zum Zeilenende werden durch Leerzeichen überschrieben (gelöscht). Wenn der Cursor bereits am Zeilenanfang steht, hat dieses Steuerzeichen keine Wirkung.

Zeile löschen

SUB **1A_H** **26_D**
Die Zeile, in der sich der Cursor befindet, wird komplett mit Leerzeichen gefüllt (gelöscht). Die Cursor-Position bleibt erhalten.

MAT 85-Mode

Cursor nach unten

ESC **1B_H** **27_D**

Der Cursor wird - unter Beibehaltung der Spaltenposition - um eine Zeile nach unten bewegt. Steht der Cursor am unteren Schriftfeld-Rand, wird der Inhalt des Schriftfeldes um eine Zeile nach oben geschoben (gescrollt). Hierbei geht der Inhalt der obersten Zeile verloren. Unten wird eine Leerzeile eingefügt.

Dieses Steuerzeichen hat bei dem hier beschriebenen Interface eine geringfügig andere Wirkung als beim Video-Interface BFZ/MFA 8.2: Befindet sich der Cursor beim Interface 8.2 am unteren Schriftfeld-Rand, wird die oben herausgeschobene Zeile unten wieder eingefügt.

Cursor in die Home-Position

FS **1C_H** **28_D**

Der Cursor wird in die linke obere Ecke des Schriftfeldes (Home-Position) gestellt. Hierbei wird der Bildschirm nicht gelöscht.

Cursor an den Zeilenanfang

GS **1D_H** **29_D**

Der Cursor wird an den Zeilenanfang bewegt.

3.1.2 Die Steuer-Sequenzen im MAT 85-Mode

Im MAT 85-Mode kennt das Interface zusätzlich zu den 1-Byte-Steuerzeichen weitere fünf Steuer-Sequenzen. Diese Sequenzen beginnen alle mit dem Zeichen DLE (Data Link Escape, 10_H) und bestehen aus jeweils drei Zeichen. Um eine Steuerfunktion auszulösen, müssen die Codes der entsprechenden drei Zeichen vom Mikrocomputer zum Interface übertragen werden. Die DLE-Sequenzen sind identisch mit den DLE-Sequenzen im TVI 950-Mode.

Für jede Sequenz ist im folgenden ihre Bezeichnung, ihre hexadezimale Code-Folge, ihre dezimale Code-Folge und ihre Funktion angegeben.

TVI 950-Mode einschalten

DLE DLE 1 **10_H 10_H 31_H** **16_D 16_D 49_D**

Mit dieser Sequenz kann in den TVI 950-Mode geschaltet werden. Nach der Umschaltung gelten die Steuerzeichen-Definitionen des TVI 950-Modes.

MAT 85-Mode

MAT 85-Mode einschalten

DLE DLE 2 $10_H 10_H 32_H$ $16_D 16_D 50_D$

Mit dieser Sequenz kann in den MAT 85-Mode geschaltet werden. Befindet man sich bereits im MAT 85-Mode, so wird diese Sequenz ignoriert.

Aktuellen Mode abfragen

DLE DLE ? $10_H 10_H 3F_H$ $16_D 16_D 63_D$

Mit Hilfe dieser Sequenz kann der Anwender abfragen, welcher Mode (MAT 85 oder TVI 950) momentan aktiv ist. Nach dem Empfang dieser Sequenz "antwortet" das Interface mit drei Bytes:

Antwort-Bytes	aktiver Mode
$4D_H$ 31_H $0D_H$	TVI 950-Mode
$4D_H$ 32_H $0D_H$	MAT 85-Mode

Beachten Sie hierzu bitte auch das Kapitel "Einsatz unter dem Betriebsprogramm MAT 85".

Software-Reset

DLE DLE @ $10_H 10_H 40_H$ $16_D 16_D 64_D$

Durch diese Sequenz kann ein sogenannter Software-Reset des Video-Interfaces ausgelöst werden. Das heißt, das Interface wird so eingestellt, wie es nach dem Anlegen der Betriebsspannung eingestellt sein würde.

Abfrage der Software-Version

DLE DLE V $10_H 10_H 56_H$ $16_D 16_D 86_D$

Mit Hilfe dieser Sequenz kann der Anwender die Versionsnummer der Software im EPROM der Interface-Baugruppe abfragen. Die Version ist durch zwei Ziffern gekennzeichnet (Versionsnummer). Nach dem Empfang dieser Sequenz "antwortet" das Interface mit fünf Bytes:

Beispiel für eine Antwort bei Version 1.0:

56_H 31_H $2F_H$ 30_H $0D_H$

Das erste Byte der Antwort ist immer 56_H (ASCII "V").

Das zweite Byte entspricht dem ASCII-Code der ersten Ziffer der Versionsnummer (hier "1").

Das dritte Byte ist (bei dieser Interface-Hardware) immer $2F_H$.

Das vierte Byte entspricht dem ASCII-Code der zweiten Ziffer der Versionsnummer (hier "0").

Das fünfte Byte der Antwort ist immer $0D_H$.

Beachten Sie hierzu bitte auch das Kapitel "Einsatz unter dem Betriebsprogramm MAT 85".

TVI 950-Mode

3.2 Der TVI 950-Emulations-Mode

Der TVI 950-Mode ist vorwiegend für den Einsatz unter dem Betriebssystem CP/M vorgesehen. Die Steuerzeichen dieses Modes unterscheiden sich teilweise von denen des MAT 85-Modes. Deshalb kann der TVI 950-Mode nur eingeschränkt unter dem Betriebsprogramm MAT 85 benutzt werden. Allerdings "kennt" das Interface im TVI 950-Mode mehr Steuerzeichen. So ist es in diesem Mode z.B. möglich, den Cursor durch Angabe eine Zeilen- und Spalten-Nummer an eine bestimmte Stelle des Schriftfeldes zu positionieren. Außerdem können die Zeichen auf vielfältige Art dargestellt werden: invers, unterstrichen, blinkend, doppelt hoch und doppelt breit.

Aufgrund seiner vielen Möglichkeiten werden Steuerfunktionen im TVI 950-Mode oft nicht durch einzelne Steuerzeichen, sondern durch Steuerzeichen-Folgen (Sequenzen) ausgelöst. Beginnen diese Sequenzen mit dem Zeichen ESC (Escape, Code 1B_H), so spricht man auch von einer *Escape-Sequenz*.

3.2.1 Die 1-Byte-Steuerzeichen im TVI 950-Mode

In diesem Kapitel sind die einzelnen Steuerzeichen (Control-Codes) beschrieben. Eine Beschreibung der Steuerzeichen-Folgen (Sequenzen) finden Sie im nächsten Kapitel.

Um die entsprechende Steuerfunktion auszulösen, muß der Code des Steuerzeichens vom Mikrocomputer zum Video-Interface übertragen werden. Näheres dazu entnehmen Sie bitte dem Kapitel "Einsatz unter dem Betriebssystem CP/M".

Für jedes Steuerzeichen ist im folgenden seine Bezeichnung, sein hexadezimaler Code, sein dezimaler Code und seine Funktion angegeben.

Summer

BEL	07_H	7_D
------------	-----------------------	----------------------

Durch die Ausgabe des Steuerzeichens BEL ertönt der im Interface eingebaute Summer für etwa 0,5 Sekunden.

Cursor nach links (Backspace)

BS	08_H	8_D
-----------	-----------------------	----------------------

Der Cursor wird um eine Spalte nach links bewegt. Befindet er sich am linken Schriftfeld-Rand, so bewegt er sich an das Ende der vorstehenden Zeile. Wenn der Cursor sich in der linken oberen Ecke des Schriftfeldes befindet, hat dieses Steuerzeichen keine Wirkung.

TVI 950-Mode

Tabulator

HT 09_H 9_D

Es werden soviele Leerzeichen ausgegeben, bis der Cursor auf dem nächsten *Tabulatorstop* steht. Tabulatorstops befinden sich in jeder achten Spalte. Steht der Cursor bereits auf oder hinter dem letzten Tabulatorstop einer Zeile, so bewegt er sich an den Anfang der nächsten Zeile. Falls der Cursor in der letzten Zeile stand, wird das Schriftfeld um eine Zeile nach oben geschoben. Der Inhalt der oberen Zeile geht verloren, unten wird eine Leerzeile eingefügt.

Spaltenzählung: 0,1,2,3,4,...

Tab.-Positionen: 0,8,16,24,32,40,48,56,64,72,80,88

Hinweis:

Wie beim original TVI 950-Terminal beginnt die Spaltenzählung der weiter unten beschriebenen Escape-Sequenzen "ESC =" und "ESC ?" bei 0, die Spaltenzählung für die Cursor-Positionsanzeige in der Status-Zeile aber bei 1.

Cursor nach unten, mit Scrollen (Line Feed, Zeilenvorschub)

LF 0A_H 10_D

Der Cursor wird - unter Beibehaltung der Spaltenposition - um eine Zeile nach unten bewegt. Wenn der Cursor am unteren Schriftfeld-Rand steht, wird das Schriftfeld um eine Zeile nach oben geschoben. Dabei geht der Inhalt der obersten Zeichen-Zeile verloren. Unten wird eine Leerzeile eingefügt.

Cursor nach oben

VT 0B_H 11_D

Der Cursor wird - unter Beibehaltung der Spaltenposition - um eine Zeile nach oben bewegt. Steht der Cursor am oberen Schriftfeld-Rand, wird dieses Steuerzeichen ignoriert.

Cursor nach rechts

FF 0C_H 12_D

Dieser Code bewegt den Cursor um eine Position nach rechts. Steht der Cursor am Ende einer Zeile, springt er an den Anfang der nächsten Zeile. Steht der Cursor am Ende der letzten Zeile, wird der Inhalt des Schriftfeldes nach oben geschoben (gescrollt). Der Inhalt der oberen Zeile geht dabei verloren, unten wird eine Leerzeile eingefügt.

TVI 950-Mode

Cursor an den Zeilenanfang (Carriage Return, Wagenrücklauf)

CR **0D_H** **13_D**
Der Cursor wird an den Zeilenanfang bewegt. Der Zeileninhalt bleibt unverändert.

Cursor nach unten, ohne Scrollen

SYN **16_H** **22_D**
Dieser Code bewegt den Cursor - unter Beibehaltung der Spaltenposition - um eine Zeile nach unten. Steht der Cursor in der letzten Zeile, wird dieser Code ignoriert.

Bildschirm löschen, Cursor in Home-Position

SUB **1A_H** **26_D**
Das komplette Schriftfeld wird mit Leerzeichen gefüllt (gelöscht). Der Cursor wird in die linke obere Ecke des Schriftfeldes (Home-Position) gestellt.

Cursor in die Home-Position

RS **1E_H** **30_D**
Dieser Code bewegt den Cursor in die linke obere Ecke des Schriftfeldes (Home-Position). Der Inhalt des Schriftfeldes bleibt unverändert.

3.2.2 Die Steuer-Sequenzen im TVI 950-Mode

Im TVI 950-Mode kennt das Interface zusätzlich zu den 1-Byte-Steuerzeichen auch noch Steuerzeichen-Folgen (Sequenzen). Diese Sequenzen beginnen entweder mit dem Zeichen ESC (Escape, 1B_H) oder mit dem Zeichen DLE (Data Link Escape, 10_H). Man spricht hier auch von Escape- bzw. DLE-Sequenzen. Um die jeweilige Steuerfunktion auszulösen, müssen alle Bytes einer Steuersequenz vom Mikrocomputer zum Interface übertragen werden.

Für jede Sequenz ist im folgenden ihre Bezeichnung, ihre hexadezimale Byte-Folge, ihre dezimale Byte-Folge und ihre Funktion angegeben.

Die einzelnen Sequenzen sind nach Anwendungsgebieten geordnet.

3.2.2.1 Bildschirm löschen, Cursor Home (Cursor nach links oben)

ESC *	1B _H 2A _H	27 _D 42 _D
ESC +	1B _H 2B _H	27 _D 43 _D
ESC ,	1B _H 2C _H	27 _D 44 _D
ESC :	1B _H 3A _H	27 _D 58 _D
ESC ;	1B _H 3B _H	27 _D 59 _D

Die vorstehenden fünf Escape-Sequenzen unterscheiden sich hinsichtlich ihrer Funktion bei dem hier beschriebenen Interface nicht. Durch alle Escape-Sequenzen wird das Schriftfeld mit Leerzeichen gefüllt (gelöscht) und der Cursor in die obere linke Ecke gestellt.

3.2.2.2 Escape-Sequenzen zum Editieren des Schriftfeldes

Ein Leerzeichen einfügen

ESC Q	1B _H 51 _H	27 _D 81 _D
-------	---------------------------------	---------------------------------

Mit Hilfe der Escape-Sequenz "ESC Q" kann an der Cursor-Position ein Zeichen in eine Zeile eingefügt werden: Der rechte Zeilenteil (ab Cursor-Position einschließlich bis zum rechten Rand) wird um eine Spalte nach rechts verschoben. In der entstehenden Lücke wird ein Leerzeichen eingefügt. Der Cursor steht auf diesem Leerzeichen. Das Zeichen, daß am rechten Schriftfeldrand stand, geht verloren.

Ein Zeichen löschen

ESC W	1B _H 57 _H	27 _D 87 _D
-------	---------------------------------	---------------------------------

Durch die Übertragung der Escape-Sequenz "ESC W" zum Interface, wird das Zeichen an der augenblicklichen Cursor-Position gelöscht. Alle rechts davon befindlichen Zeichen rücken um eine Spalte nach links. In der letzten Spalte der Zeile wird ein Leerzeichen eingefügt. Die Cursor-Position bleibt unverändert.

Eine Zeile einfügen

ESC E	1B _H 45 _H	27 _D 69 _D
-------	---------------------------------	---------------------------------

Alle Zeilen einschließlich der aktuellen Cursor-Zeile abwärts werden um eine Zeile nach unten verschoben. Dadurch entsteht eine Leerzeile, in deren erster Spalte der Cursor steht. Die Zeile, die am unteren Schriftfeldrand steht, fällt fort.

TVI 950-Mode

Eine Zeile löschen

ESC R **1B_H 52_H** **27_D 82_D**

Dieser Befehl löscht die aktuelle Cursor-Zeile und rückt alle weiteren Zeilen von unten auf. Am unteren Schriftfeldrand wird eine Leerzeile eingefügt. Der Cursor behält seine Zeilen-Position bei. Er wird aber an den Anfang dieser Zeile gestellt.

Zeile ab Cursor-Position löschen

ESC T **1B_H 54_H** **27_D 84_D**

ESC t **1B_H 74_H** **27_D 116_D**

Durch jeweils eine der beiden oben aufgeführten Escape-Sequenzen wird die Zeile ab der Cursor-Position (einschließlich) bis zum Zeilenende gelöscht. Der restliche Schriftfeld-Inhalt und die Cursor-Position bleiben unverändert.

Schriftfeld ab Cursor-Position löschen

ESC Y **1B_H 59_H** **27_D 89_D**

ESC y **1B_H 79_H** **27_D 121_D**

Durch jeweils eine der beiden oben aufgeführten Escape-Sequenzen wird das Schriftfeld ab der Cursor-Position (einschließlich) bis zur unteren rechten Ecke gelöscht. Die Cursor-Position ändert sich nicht.

3.2.2.3 Cursor-Positionierung**Cursor-Position setzen**

ESC = Reihe+20_H Spalte+20_H 1B_H 3D_H Reihe+20_H Spalte+20_H 27_D 61_D Reihe+32_D Spalte+32_D

Mit Hilfe dieser Escape-Sequenz kann der Cursor auf eine beliebige Position des Schriftfeldes gesetzt werden. Die für diese Escape-Sequenz gültige Reihen- und Spaltenzählung beginnt jeweils bei 0 (siehe auch unten: "Hinweis"). Deshalb hat die linke obere Ecke die Koordinaten 0,0 (Reihe, Spalte). Die Maximalwerte für Reihen- und Spaltenposition hängen von der eingestellten Reihen- (Zeilen-) und Spaltenanzahl ab.

Wertebereiche:

Zeilenanzahl	zul. Reihenposition
22	0...21
24	0...23
26	0...25
28	0...27

TVI 950-Mode

Spaltenanzahl	zul. Spaltenposition
72	0...71
80	0...79
88	0...87
96	0...95

Werden zu große Werte für Reihe oder Spalte übertragen, so bewegt sich der Cursor zum unteren bzw. zum rechten Schriftfeldrand.

Wie allgemein üblich, wird auch bei diesem Interface die Reihen- und Spaltenposition nicht direkt angegeben. Stattdessen wird auf die Positionsangaben jeweils ein Offset von 20_H (32_D) aufaddiert. So werden als Positionsangaben keine Steuerzeichen übertragen.

Beispiel:

Der Cursor soll in die dritte Reihe der ersten Spalte gesetzt werden. Da die Reihen und Spalten jeweils ab Null gezählt werden, ist dies die Position 2,0. Zu diesen Werten muß noch jeweils der Offset 20_H addiert werden. Dies ergibt: $22_H, 20_H$. Die komplette Steuerzeichenfolge, die zur Positionierung des Cursors erforderlich ist, lautet also: $1B_H 3D_H 22_H 20_H$.

Hinweis:

Wie beim original TVI 950-Terminal beginnt die Spalten- und Reihenzählung zur Positionierung des Cursors durch "ESC =" bei 0, die Zählung für die Cursor-Positionsanzeige in der Status-Zeile aber bei 1.

Cursor-Position abfragen

ESC ?

$1B_H 3F_H$

$27_D 63_D$

Diese Escape-Sequenz ermöglicht es dem Anwender, die aktuelle Cursor-Position abzufragen. Die für diese Escape-Sequenz gültige Reihen- und Spaltenzählung beginnt jeweils bei 0 (siehe auch unten: "Hinweis"). Das Video-Interface sendet die Reihen- und Spaltenposition jeweils mit einem Offset von 20_H an den Mikrocomputer. Das letzte Zeichen ist immer $0D_H$ (Carriage Return).

TVI 950-Mode

Beispiel:

Der Cursor befindet sich in der dritten Reihe der ersten Spalte. Da die Reihen und Spalten jeweils ab Null gezählt werden, ist dies die Position 2,0. Überträgt der Anwender nun die Sequenz "ESC ?" zum Video-Interface, so sendet dieses die aktuelle Cursor-Position als "Antwort" zum Mikrocomputer. Bei der angenommenen Position würden folgende Bytes als "Antwort" gesendet:

$$22_{\text{H}} \ 20_{\text{H}} \ 0\text{D}_{\text{H}}$$

Das erste Byte ist die Reihen-Position + 20_{H} .

Das zweite Byte ist die Spalten-Position + 20_{H} .

Das dritte Byte ist immer 0D_{H} .

Hinweis:

Wie beim original TVI 950-Terminal beginnt die Spalten- und Reihenzählung zur Abfrage der Cursor-Position mit Hilfe von "ESC ?" bei 0, die Zählung für die Cursor-Positionsanzeige in der Status-Zeile aber bei 1.

Cursor nach links zum nächsten Tabulator (Backtab)
ESC I $1\text{B}_{\text{H}} \ 49_{\text{H}}$ $27_{\text{D}} \ 73_{\text{D}}$

Mit Hilfe dieser Escape-Sequenz wird der Cursor zurück (nach links) zum nächsten Tabulator bewegt. Dies wird auch als "Backtab-Funktion" bezeichnet. Befindet sich der Cursor am Zeilenanfang, wird er auf den letzten Tabulator der vorigen Zeile gesetzt. Befindet er sich in der linken oberen Ecke (Home-Position), wird der Befehl ignoriert.

3.2.2.4 Zeichensätze, Zeichendarstellung (Attribute), Cursor-Darstellung**Zeichensatz wählen****ESC z Auswahlcode** $1\text{B}_{\text{H}} \ 7\text{A}_{\text{H}} \ \text{Auswahlcode}$ $27_{\text{D}} \ 122_{\text{D}} \ \text{Auswahlcode}$

Mit der hier aufgeführten Escape-Sequenz kann einer von acht länderspezifischen Zeichensätzen gewählt werden. Diese Zeichensätze unterscheiden sich nur bei den Zeichencodes 23_{H} , 24_{H} , 40_{H} , 5B_{H} , 5C_{H} , 5D_{H} , 5E_{H} , 60_{H} , 7B_{H} , 7C_{H} , 7D_{H} und 7E_{H} . Die den einzelnen Zeichensätzen zugeordneten Zeichen können dem Anhang entnommen werden.

Um einen bestimmten Zeichensatz zu wählen, müssen drei Bytes zum Video-Interface übertragen werden:

1. ESC (1B_{H} , 27_{D})
2. z (7A_{H} , 122_{D})
3. Auswahlcode (siehe Tabelle)

TVI 950-Mode

Der Auswahlcode bestimmt den gewählten Zeichensatz. Die folgende Tabelle zeigt die Zuordnung:

Zeichensatz	Auswahlcode		
	ASCII	hex.	dez.
USA	0	30	48
Frankreich	1	31	49
Deutschland	2	32	50
England	3	33	51
Dänemark	4	34	52
Schweden	5	35	53
Italien	6	36	54
Spanien	7	37	55

Der neue Zeichensatz wirkt sich nicht auf die Zeichen aus, die bereits vor der Zeichensatz-Umschaltung auf dem Bildschirm dargestellt wurden. Er gilt nur für die Zeichen, die **nach** der Wahl des neuen Zeichensatzes zum Interface übertragen werden.

Auf der Tastatur befinden sich nur die Symbole des USA-Zeichensatzes. Die nachfolgende Tabelle zeigt, welche Tasten zu betätigen sind, um die entsprechenden länderspezifischen Zeichen einzugeben:

Taste	Länderzeichensätze						
	FRA	GER	ENG	DEN	SWE	ITA	SPA
#	#	#	£	#	#	#	#
\$	\$	\$	\$	\$	¤	\$	\$
@	à	§	@	@	@	@	@
[°	Ä	[Æ	Ä	°	ì
\	ç	Ö	\	ø	Ö	\	ñ
]	§	Ü]	Å	Å	é]
^	^	^	^	^	Ü	^	^
`	`	`	`	`	é	ù	`
{	é	ä	{	æ	ä	à	..
	ù	ö		ø	ö	ö	ñ
}	è	ü	}	å	å	è	}
~	..	ß	~	~	ü	ì	~

TVI 950-Mode

Attribute setzen/löschen

ESC G Attribute+30_H 1B_H 47_H Attribute+30_H 27_D 71_D Attribute+48_D

Die auf dem Bildschirm darzustellenden Zeichen können auf verschiedene Arten angezeigt werden. Soll von der Normaldarstellung abgewichen werden, sind entsprechende Attribute einzuschalten. Folgende Attribute sind möglich:

unsichtbar
 blinkend
 invers
 unterstrichen
 doppelt breit
 doppelt hoch

Dabei können fast alle Attribute miteinander kombiniert werden. Lediglich bei den Attributen "unsichtbar" und "doppelt hoch" ist eine Kombination miteinander nicht möglich.

Um Attribute ein- oder auszuschalten müssen jeweils drei Bytes zum Interface übertragen werden:

1. ESC (1B_H, 27_D)
2. G (47_H, 71_D)
3. Attribute + 30_H (s.u.)

Das dritte Byte "Attribute + 30_H" gibt die gewünschten Zustände (Ein/Aus) der Attribute an. Es können beliebig viele Attribute gleichzeitig ein- bzw. ausgeschaltet werden.

Eine Änderung der Attribut-Zustände wirkt sich auf alle Zeichen aus, die nach der Änderung zum Interface übertragen werden.

Das Byte "Attribute + 30_H" ergibt sich durch das Setzen bzw. Löschen bestimmter Bits (zum Ein- bzw. Ausschalten bestimmter Attribute) und durch die anschließende Addition von 30_H. Welche Bits zum Einschalten bestimmter Attribute gesetzt werden müssen, kann der folgenden Aufstellung entnommen werden:

Attribut	Bit
unsichtbar	0
blinkend	1
invers	2
unterstrichen	3
doppelt breit	4
doppelt hoch	5

Alle anderen Bits sind reserviert und dürfen nicht gesetzt werden.

TVI 950-Mode

Beispiel:

Die nachfolgenden Zeichen sollen blinkend und unterstrichen dargestellt werden. Dazu sind die Attribute "blinkend" und "unterstrichen" ein- und alle anderen Attribute auszuschalten. Es ergibt sich das folgende Bitmuster:

Bit-Nummer	7	6	5	4	3	2	1	0
Bitmuster	0	0	0	0	1	0	1	0
	0				A			

Aus dem Bitmuster ergibt sich der Wert $0A_H$. Zu diesem Wert muß nun noch der Offset 30_H addiert werden. Man erhält dann den Wert $3A_H$. Die vollständige Sequenz, die zum Einschalten der gewünschten Attribute zum Interface übertragen werden muß, lautet also: $1B_H$ 47_H $3A_H$. Durch die Übertragung dieser Byte-Folge werden die Attribute "unsichtbar", "invers", "doppelt breit" und "doppelt hoch" ausgeschaltet, da die entsprechenden Bits im Bitmuster "0" sind.

Im Anhang finden Sie eine Tabelle, die Ihnen die Ermittlung des Bytes "Attribute + 30_H " erleichtert.

Mit der Escape-Sequenz "ESC G 0" ($1B_H$ 47_H 30_H) können alle Attribute gleichzeitig ausgeschaltet werden.

Besondere Hinweise:

Bei der Verwendung der Attribute "doppelt hoch" und "doppelt breit" ist zu beachten, daß sich die meisten Steuerbefehle auf Zeichen einfacher Höhe und Breite beziehen. Bei Zeichen mit doppelter Höhe und/oder Breite können sie entweder keine oder aber eine andere Wirkung als vorgesehen haben. Der einzige Steuercode, der die Schriftgröße berücksichtigt, ist "Line Feed" (Zeilenvorschub, $0A_H$). Falls das Attribut "doppelt hoch" aktiv ist, werden zwei Standard-Zeilenvorschübe ausgeführt. Dadurch entsteht bei Abschluß jeder Zeile mit CR/LF ein brauchbares Schriftbild. Bei Zeilenüberlauf wird auch bei doppelter Zeichenhöhe nur ein Zeilenvorschub ausgeführt, wodurch die untere Hälfte der vorigen Zeile von der nächsten überschrieben wird.

Bei doppelt hoher Schrift werden die einzelnen Zeichen grundsätzlich nach oben erweitert. Sie überschreiben dabei eventuell Zeichen, die in der darüberliegenden Zeile stehen. Steht der Cursor in der obersten Zeile (z.B. Home-Position: 0,0) so wird der Cursor nach Übertragung des ersten Zeichens aber vor der Zeichendarstellung vom Video-Interface eine Zeile tiefer positioniert. Dadurch lassen sich auch nach einem Cursor-Home-Befehl sofort doppelt hohe Zeichen darstellen.

Bei doppelt breiter Schrift kann es zu Verzerrungen kommen, wenn doppelt breite Schrift auf einer ungeraden Spalte beginnt (Zählung ab 0). In diesem Fall werden beim Zeilenüberlauf in der nächsten Zeile jeweils nicht zusammenhängende Buchstabenhälften angezeigt.

TVI 950-Mode

Halbe Helligkeit

ESC) $1B_H 29_H$ $27_D 41_D$

Das Interface kann die Zeichen in zwei Helligkeitsstufen anzeigen. Mit der hier aufgeführten Escape-Sequenz wird für alle nachfolgend übertragenen Zeichen die halbe Helligkeit gewählt. Normaleinstellung ist die volle Helligkeit. "ESC (" schaltet wieder auf die volle Helligkeit zurück.

Volle Helligkeit

ESC ($1B_H 28_H$ $27_D 40_D$

Mit der hier aufgeführten Escape-Sequenz kann die Zeichendarstellung auf die volle Helligkeit (Normaleinstellung) geschaltet werden. Siehe auch "ESC)".

Bildhintergrund hell (bright)

Normaldarstellung als "dunkle Zeichen auf hellem Grund"

ESC b $1B_H 62_H$ $27_D 98_D$

Mit dieser Escape-Sequenz wird für alle Zeichen ohne das Attribut "invers" die Darstellungsart "dunkle Zeichen auf hellem Grund" gewählt. Für Zeichen mit dem Attribut "invers" gilt: helle Zeichen auf dunklem Grund. Dies ist die Grundeinstellung, wenn über den Schalter S1.4 ein heller Bildhintergrund gewählt wurde. Diese Escape-Sequenz wirkt sich auf alle Zeichen aus (auch auf die bereits angezeigten). Siehe auch "ESC d".

Bildhintergrund dunkel (dark)

Normaldarstellung als "helle Zeichen auf dunklem Grund"

ESC d $1B_H 64_H$ $27_D 100_D$

Mit dieser Escape-Sequenz wird für alle Zeichen ohne das Attribut "invers" die Darstellungsart "helle Zeichen auf dunklem Grund" gewählt. Für Zeichen mit dem Attribut "invers" gilt: dunkle Zeichen auf hellem Grund. Dies ist die Grundeinstellung, wenn über den Schalter S1.4 ein dunkler Bildhintergrund gewählt wurde. Diese Escape-Sequenz wirkt sich auf alle Zeichen aus (auch auf die bereits angezeigten). Siehe auch: "ESC b".

Control-Mode einschalten

ESC U $1B_H 55_H$ $27_D 85_D$

Mit dieser Escape-Sequenz wird der "Control-Mode" eingeschaltet. In diesem Mode werden alle Steuerzeichen, die das Video-Interface empfängt, nicht ausgeführt, sondern als Control-Zeichen angezeigt. Siehe auch das Kapitel "Der Control-Mode".

TVI 950-Mode

Beispiel:

Wird das Steuerzeichen BEL (07_H) bei eingeschaltetem Control-Mode vom Mikrocomputer zum Interface übertragen, so ertönt **nicht** der Summer des Interfaces. Stattdessen wird an der Cursor-Position ein inverses "G" mit halber Helligkeit dargestellt. Das "G" steht für die Tastenkombination "Control G", die den Code 07_H erzeugt.

Hinweis:

Da alle Steuerzeichen nur noch angezeigt aber nicht mehr ausgeführt werden, kann der Control-Mode nur über das Set-Up-Menü (Control Mode AUS) verlassen werden. Siehe Kapitel "Das Set-Up-Menü".

Cursor-Darstellung

ESC . Darstellung

$1B_H$ $2E_H$ Darstellung

27_D 46_D Darstellung

Mit Hilfe dieser Escape-Sequenz kann die Art der Cursor-Darstellung den Wünschen des Anwenders angepaßt werden. Hierzu sind drei Bytes zum Interface zu übertragen:

1. ESC ($1B_H$, 27_D)
2. . ($2E_H$, 46_D)
3. Darstellung (siehe Tabelle)

Das dritte Byte ("Darstellung") bestimmt die Darstellungsart des Cursors. Welches "Darstellung"-Byte zum Interface übertragen werden muß, kann der folgenden Tabelle entnommen werden:

Darstellungsart	"Darstellung"		
	ASCII	hex.	dez.
keinen Cursor anzeigen	0	30_H	48_D
blinkender Block	1	31_H	49_D
stehender Block	2	32_H	50_D
blinkender Unterstrich	3	33_H	51_D
stehender Unterstrich	4	34_H	52_D

Beispiel:

Soll für den Cursor die Darstellungsart "blinkender Unterstrich" gewählt werden, so ist die Byte-Folge $1B_H$ $2E_H$ 33_H zum Interface zu übertragen.

TVI 950-Mode

3.2.2.5 Die DLE-Sequenzen im TVI 950-Mode

Die DLE-Sequenzen sind identisch mit den DLE-Sequenzen im MAT 85-Mode.

TVI 950-Mode einschalten

DLE DLE 1 $10_H 10_H 31_H$ $16_D 16_D 49_D$

Mit dieser Sequenz kann in den TVI 950-Mode umgeschaltet werden. Befindet man sich bereits im TVI 950-Mode, so wird diese Sequenz ignoriert.

MAT 85-Mode einschalten

DLE DLE 2 $10_H 10_H 32_H$ $16_D 16_D 50_D$

Mit dieser Sequenz kann in den MAT 85-Mode geschaltet werden. Nach der Umschaltung gelten die Steuerzeichen-Definitionen des MAT 85-Modus.

Aktuellen Mode abfragen

DLE DLE ? $10_H 10_H 3F_H$ $16_D 16_D 63_D$

Mit Hilfe dieser Sequenz kann der Anwender abfragen, welcher Mode (MAT 85 oder TVI 950) momentan aktiv ist. Nach dem Empfang dieser Sequenz "antwortet" das Interface mit drei Bytes:

Antwort-Bytes			aktiver Mode
$4D_H$	31_H	$0D_H$	TVI 950-Mode
$4D_H$	32_H	$0D_H$	MAT 85-Mode

Beachten Sie hierzu bitte auch das Kapitel "Einsatz unter dem Betriebssystem CP/M".

Software-Reset

DLE DLE @ $10_H 10_H 40_H$ $16_D 16_D 64_D$

Durch diese Sequenz kann ein sogenannter Software-Reset des Video-Interfaces ausgelöst werden. Das heißt, das Interface wird so eingestellt, wie es nach dem Anlegen der Betriebsspannung eingestellt sein würde.

TVI 950-Mode

Abfrage der Software-Version

DLE DLE V $10_H 10_H 56_H$ $16_D 16_D 86_D$

Mit Hilfe dieser Sequenz kann der Anwender die Versionsnummer der Software im EPROM der Interface-Baugruppe abfragen. Die Version ist durch zwei Ziffern gekennzeichnet (Versions-Nummer). Nach dem Empfang dieser Sequenz "antwortet" das Interface mit fünf Bytes:

Beispiel für eine Antwort bei Version 1.0:

56_H 31_H $2F_H$ 30_H $0D_H$

Das erste Byte der Antwort ist immer 56_H (ASCII "V").

Das zweite Byte entspricht dem ASCII-Code der ersten Ziffer der Versions-Nummer (hier "1").

Das dritte Byte ist (bei dieser Interface-Hardware) immer $2F_H$.

Das vierte Byte entspricht dem ASCII-Code der zweiten Ziffer der Versions-Nummer (hier "0").

Das fünfte Byte der Antwort ist immer $0D_H$.

Beachten Sie hierzu bitte auch das Kapitel "Einsatz unter dem Betriebssystem CP/M".

4 Einsatz unter dem Betriebsprogramm MAT 85

Für den Einsatz unter dem Betriebsprogramm MAT 85 des MFA-Mikrocomputers ist der MAT 85-Mode des Video-Interfaces vorgesehen. Der Datenaustausch zwischen CPU-Baugruppe und Video-Interface erfolgt über die Systembus-Leitungen Tx und Rx (siehe Bild 7).

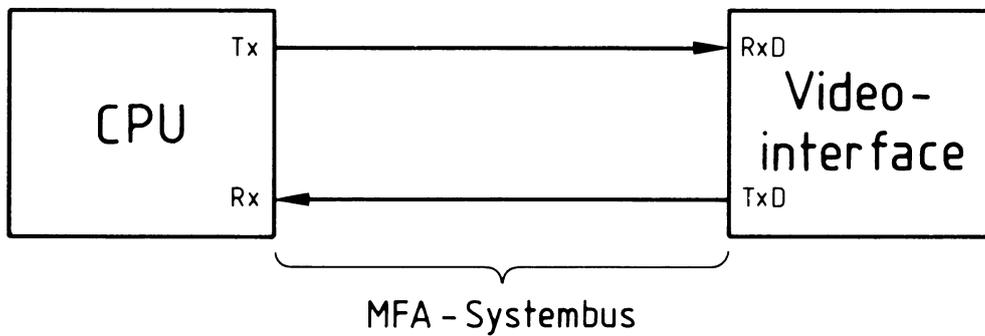


Bild 7: Datenaustausch zwischen CPU und Video-Interface (MAT 85)

Vor der Inbetriebnahme sind die Schalter der Interface-Baugruppe entsprechend dem nachfolgenden Kapitel "Schalterstellungen" einzustellen.

Die Kapitel "Nutzung in Assembler Programmen" und "Nutzung unter BFZ-STEUER-BASIC" enthalten Informationen für Anwender, welche die Steuerfunktionen des Video-Interfaces in eigenen Assembler- bzw. BASIC-Programmen nutzen wollen.

An dieser Stelle soll darauf hingewiesen werden, daß sowohl der MAT 85-Assembler als auch das BFZ-STEUER-BASIC bei der Eingabe von Programmzeilen alle Kleinbuchstaben in Großbuchstaben umwandeln. So können diese Programme die eingegebenen Assembler- bzw. BASIC-Anweisungen einfacher erkennen.

Beachten Sie bitte, daß nach dem Einschalten des Mikrocomputer-Systems die Leertaste betätigt werden muß. Dies gilt auch dann, wenn auf dem Bildschirm keine Zeichen angezeigt werden.

Einsatz unter MAT 85

4.1 Schalterstellungen

Auf der Interface-Baugruppe befinden sich drei Schalter-Reihen (siehe auch Bestückungsplan im Anhang) zur Grundeinstellung des Video-Interfaces beim Einschalten. Jede der mit S1, S2 und S3 bezeichneten Schalter-Reihen besteht aus acht Schaltern. S1.1 bezeichnet den Schalter Nummer 1 in der Schalter-Reihe S1. Die anderen Schalter sind entsprechend numeriert. Für das Betriebsprogramm MAT 85 sind die Schalter folgendermaßen einzustellen:

- Schalter-Reihe S1:

Schalter	Stellung	eingestellte Funktion
S1.1	ON	MAT 85-Mode
S1.2	OFF	blinkender Cursor
S1.3	OFF	Strich-Cursor
S1.4	OFF	dunkler Bildhintergrund
S1.5	OFF	reserviert, immer OFF!
S1.6	OFF	DTR-Handshake
S1.7	OFF	reserviert, immer OFF!
S1.8	OFF	Statuszeile aus

- Schalter-Reihe S2

Schalter	Stellung	eingestellte Funktion
S2.1	OFF	Wortlänge 7 Bit
S2.2	OFF	Zeichensatz USA ¹⁾
S2.3	OFF	dto. ¹⁾
S2.4	OFF	dto. ¹⁾
S2.5	ON	Zeilenanzahl 24
S2.6	OFF	dto.
S2.7	ON	Spaltenanzahl 80
S2.8	OFF	dto.

¹⁾ Entspricht dem Zeichensatz des Video-Interfaces BFZ/MFA 8.2

Einsatz unter MAT 85

- Schalter-Reihe S3

Schalter	Stellung	eingestellte Funktion
S3.1	ON	2 Stopbits
S3.2	ON	Mark Parity
S3.3	OFF	dto.
S3.4	ON	dto.
S3.5	ON	1200 Baud
S3.6	OFF	dto.
S3.7	ON	dto.
S3.8	OFF	dto.

Anmerkung:

Die Schalter S1.2 (Cursor-Art), S1.3 (Cursor-Form), S1.4 (Bildhintergrund), S1.8 (Statuszeile) und S2.2, S2.3, S2.4 (alle drei zur Einstellung des Zeichensatzes) können - den persönlichen Wünschen entsprechend - anders eingestellt werden. Eine andere Einstellung **dieser** Schalter beeinträchtigt die Funktion nicht. Angaben über die durch diese Schalter einstellbaren Optionen entnehmen Sie bitte dem Anhang.

4.2 Nutzung in Assembler-Programmen**4.2.1 Zeichenausgabe**

Da unter dem Betriebsprogramm MAT 85 der MAT 85-Mode des Interfaces genutzt wird, können grundsätzlich alle Ausgabe-Routinen, die in der MAT 85-Beschreibung dokumentiert sind, verwendet werden. Hier sollen nur einige dieser Routinen genannt werden:

WCHAR, Adresse 0052

Sendet das Zeichen, dessen Code sich im Akkumulator befindet, zum Interface

```

MVI A,0D      ;CODE FUER WAGENRUECKLAUF
CALL 0052    ;CODE AUSGEBEN
MVI A,'X'    ;CODE FUER 'X'
CALL 0052    ;CODE AUSGEBEN
JMP 0040     ;ENDE, ZURUECK ZUR MAT 85-KOMMANDOEINGABE

```

Einsatz unter MAT 85

WCHARI, Adresse 0055

Gibt 1 Zeichen, dessen Code nach dem CALL-Befehl im Speicher steht, aus.

```
CALL 0055 ;NACHFOLGENDEN CODE AUSGEBEN
DB   0D   ;CODE FUER WAGENRUECKLAUF
CALL 0055 ;NACHFOLGENDEN CODE AUSGEBEN
DB   'X'  ;CODE FUER 'X'
JMP  0040 ;ENDE, ZURUECK ZUR MAT 85-KOMMANDOEINGABE
```

WBUFI, Adresse 006D

Gibt den hinter dem CALL-Befehl stehenden Text auf dem Bildschirm aus. Am Ende des Textes muß als Endemarke das Byte 00_H stehen.

```
CALL 006D ;NACHFOLGENDE CODES AUSGEBEN
DB   0D, 'X';CODES FUER WAGENRUECKLAUF UND 'X'
DB   00   ;TEXT-ENDEMARKE
JMP  0040 ;ENDE, ZURUECK ZUR MAT 85-KOMMANDOEINGABE
```

WBUF, Adresse 0BA1

Gibt Text aus einem Textpuffer aus, dessen Anfangsadresse sich vor dem Aufruf der Routine im HL-Registerpaar befinden muß. Am Ende des Textes muß als Endemarke 00_H stehen.

```
LXI  H, BUF ;ANFANGSADRESSE DES PUFFERS
CALL 0BA1 ;CODES AUSGEBEN
JMP  0040 ;ENDE, ZURUECK ZUR MAT 85-KOMMANDOEINGABE
BUF: DB   0D, 'X';AUSZUGEBENDE CODES
     DB   00   ;TEXT-ENDEMARKE
```

4.2.2 Zeicheneingabe

Alle vom Interface gesendeten Zeichen - gleichgültig, ob sie über die Tastatur eingegeben wurden oder die "Antwort" einer DLE-Sequenz sind - können unter Verwendung des MAT 85-Unterprogramms "RCHAR" (Adresse 0043_H) eingelesen werden.

Beispiel:

Das nachstehende Programm gibt im ersten Teil die Steuersequenz "DLE DLE ?" zur Abfrage des aktuellen Modes aus. Die aus 3 Bytes bestehende Antwort des Terminals wird in einen Puffer eingelesen und anschließend angezeigt. Die Anzeige kann nicht sofort nach dem Empfang eines Antwort-Bytes erfolgen, da sonst die folgenden Bytes verloren gehen.

Einsatz unter MAT 85

Im zweiten Teil des Programms werden in einer Endlosschleife Zeichen vom Video-Interface gelesen und sofort zum Video-Interface zurückgesendet. Dies gilt auch für Steuerzeichen. Durch die Betätigung der CR-Taste (Code $0D_H$) wird das Programm beendet.

Begründet durch die verwendete Routine "RCHAR" des MAT 85-Betriebsprogramms, kann das Programm auch durch die Betätigung der ESC-Taste abgebrochen werden.

```

START:   LXI  H,MODE      ;ZEIGER AUF STEUERZ. FUER MODEABFRAGE
         CALL OBA1       ;GEBE STEUERZEICHEN AUS
         LXI  H,BUF      ;PUFFER FUER DIE ANTWORT-ZEICHEN
         MVI  B,3        ;ZAEHLER: DREI ZEICHEN EINLESEN
LOOP:    CALL 0043       ;LESE ZEICHEN IN DEN AKKUMULATOR
         MOV  M,A        ;SCHREIBE ZEICHEN IN DEN PUFFER
         INX  H          ;ZEIGER WEITERSTELLEN
         DCR  B          ;ANTWORT KOMPLETT GELESEN ?
         JNZ  LOOP       ;NEIN --> LOOP
         LXI  H,BUF      ;ZEIGER AUF 1. ANTWORT-ZEICHEN
         CALL OBA1       ;ANTWORT AUF DEM BILDSCHIRM ANZEIGEN
;
TEIL2:   CALL 0043       ;LESE ZEICHEN IN DEN AKKUMULATOR
         CPI  0D         ;CARRIAGE-RETURN ?
         JZ   0040       ;JA --> 0040 (ENDE)
         CALL 0052       ;NEIN (NICHT 0D) --> ZEIGE ZEICHEN AN
         JMP  TEIL2      ;LESE WEITERE ZEICHEN
;
MODE:    DB   10,10,'?' ;STEUERZEICHEN F. MODEABFRAGE
         DB   00        ;ENDEMARKE
BUF:     DB   00,00,00 ;BUFFER FUER ANTWORT
         DB   00        ;ENDEMARKE

```

4.2.3 Ein Programmbeispiel in Assembler-Sprache

Das nachstehende Programm fordert den Anwender auf, eine Ziffer einzugeben. Der Aufforderungstext wird mit Hilfe des Unterprogramms "TXTAUS" (MAT 85-Routine $OBA1_H$) ausgegeben. Über die Routine 0043_H wird die Tastatur abgefragt. Das Programm überprüft, ob tatsächlich eine Ziffer (ASCII-Code 30_H bis 39_H) eingegeben wurde. Ist dies der Fall, gibt das Programm die entsprechende Ziffer als Wort auf dem Bildschirm aus. Wurde keine Ziffer eingegeben, wird die Meldung "KEINE ZIFFER" angezeigt. Das hier beschriebene Programm wertet die Betätigung der "CR"-Taste als Abbruchwunsch aus. Ein Abbruch ist außerdem durch Betätigung der "ESC"-Taste möglich. Dies ergibt sich durch die verwendete MAT 85-Routine 0043_H .

Einsatz unter MAT 85

```

START:   LXI  H,TEXT1      ;ZEIGER AUF 1. TEXT
         CALL OBA1        ;TEXT AUSGEBEN
         CALL 0043        ;ZEICHEN VON TASTATUR EINLESEN
         CPI  0D          ;CR-TASTE BETAETIGT ?
         JZ   0040        ;WENN JA --> 0040H, ENDE
         CPI  30          ;ASCII-CODE < 30H ?
         JC   NOZIFF      ;WENN JA --> NOZIFF
         CPI  3A          ;ASCII-CODE >= 3AH ?
         JNC NOZIFF      ;WENN JA --> NOZIFF
         SUI  2F          ;ASCII-CODE AUF 1...10 (DEZ) REDUZIEREN
         MOV  B,A         ;ERMITTELTEN WERT (ZAEHLER) NACH B
         LXI  H,ZIFF      ;ZEIGER AUF ZIFFERN-TEXTE
SUCHO:   MOV  A,M         ;ZEICHEN LESEN
         INX  H           ;ZEIGER AUF NAECHSTES ZEICHEN STELLEN
         ORA  A           ;00H (TRENNZEICHEN) ?
         JNZ  SUCHO      ;WENN NEIN --> SUCHO
         DCR  B           ;ZAEHLER MINUS EINS
         JNZ  SUCHO      ;ZAEHLER NICHT NULL --> SUCHO
         CALL OBA1        ;TEXT AUSGEBEN
         JMP  START      ;AUF EIN NEUES
;-----
NOZIFF:  LXI  H,NZTXT     ;ZEIGER AUF FEHLERMELDUNG
         CALL OBA1        ;MELDUNG AUSGEBEN
         JMP  START      ;AUF EIN NEUES
;-----
TEXT1:   DB   0D,0A       ;WAGENRUECKLAUF, ZEILENVORSCHUB
         DB   'BITTE EINE ZIFFER EINGEBEN'
         DB   0D,0A       ;WAGENRUECKLAUF, ZEILENVORSCHUB
         DB   00          ;ENDEMARKE
NZTXT:   DB   07          ;BELL
         DB   'KEINE ZIFFER'
         DB   0D,0A       ;WAGENRUECKLAUF, ZEILENVORSCHUB
         DB   00          ;ENDEMARKE
ZIFF:    DB   00          ;MARKE
         DB   'NULL'
         DB   00          ;MARKE
         DB   'EINS'
         DB   00          ;MARKE
         DB   'ZWEI'
         DB   00          ;MARKE
         DB   'DREI'
         DB   00          ;MARKE
         DB   'VIER'

```

Einsatz unter MAT 85

```
DB 00 ;MARKE
DB 'FUENF'
DB 00 ;MARKE
DB 'SECHS'
DB 00 ;MARKE
DB 'SIEBEN'
DB 00 ;MARKE
DB 'ACHT'
DB 00 ;MARKE
DB 'NEUN'
DB 00 ;MARKE
```

4.3 Nutzung unter BFZ-STEUER-BASIC

4.3.1 Zeichenausgabe

Darstellbare Zeichen (wie Buchstaben und Ziffern) können unter BFZ-STEUER-BASIC mit Hilfe der PRINT-Anweisung ausgegeben werden:

```
10 PRINT "Video-Interface"
```

Für die Ausgabe von Steuerzeichen ist ein kleines Hilfsprogramm erforderlich. Es kann mit dem nachstehenden BASIC-Programm in den RAM-Speicher eingetragen werden:

```
32000 POKE DEC(E000),125
32010 POKE DEC(E001),205
32020 POKE DEC(E002),82
32030 POKE DEC(E003),0
32040 POKE DEC(E004),201
```

Das Hilfsprogramm besteht aus drei Assembler-Anweisungen:

```
E000 MOV A,L ; Lade Ausgabe-Zeichen vom L-Register in den Akku
E001 CALL 0052 ; Gebe Zeichen aus
E004 RET ; Zurück zum BASIC-Programm
```

Das Hilfsprogramm ermöglicht die Ausgabe aller ASCII-Zeichen. Dazu wird es über die USR-Funktion aufgerufen. Alle vier nachstehend aufgeführten Programm-Zeilen sind gleichwertig und zeigen verschiedene Arten des USR-Aufrufs. In allen vier Fällen wird das Steuerzeichen VT (Cursor Up - Cursor hoch - Code=0B_H (11 dez.)) ausgegeben. Vergleiche auch: Anleitung zum BFZ-STEUER-BASIC in den Fachtheoretischen Übungen (FTÜ).

Einsatz unter MAT 85

```

100 X=USR(DEC(E000),DEC(0B)) : REM --- GIBT STEUERZEICHEN OB AUS
110 X=USR(DEC(E000),11) : REM --- WIE OBEN
120 C=DEC(0B) : X=USR(DEC(E000,C)) : REM --- WIE OBEN
130 C=11 : X=USR(DEC(E000,C)) : REM --- WIE OBEN

```

Bitte beachten Sie, daß das Hilfsprogramm vor seinem ersten Aufruf im RAM installiert worden sein muß! Dazu können die oben aufgeführten BASIC-Programmzeilen 32000 bis 32040 als Unterprogramm in Ihr Programm eingefügt und zu Beginn aufgerufen werden:

```

1 GOSUB 32000 : REM --- HILFSROUTINE IN DAS RAM LADEN
:
:
111 X=USR(DEC(E000),C) : REM --- ZEICHEN (CODE IN C) AUSGEBEN
:
:
31999 END
32000 POKE DEC(E000),125
32010 POKE DEC(E001),205
32020 POKE DEC(E002),82
32030 POKE DEC(E003),0
32040 POKE DEC(E004),201
32050 RETURN

```

4.3.2 Zeicheneingabe

Tastatureingaben können mit Hilfe der INPUT- oder der \$-Anweisung eingelesen werden. Genaueres über diese Anweisungen können Sie der Anleitung zum BFZ-STEUER-BASIC im FTÜ-Band entnehmen.

Beispiele:

```

10 INPUT "GEWICHT (KG) ",G
20 PRINT "DAS GEWICHT BETRAEGT";G;"KILOGRAMM"
30 PRINT "BITTE BETAETIGEN SIE EINE TASTE"
40 T=$
50 PRINT "DIE BETAETIGTE TASTE HAT DEN CODE";T;"(DEZ.)"

```

Die DLE-Sequenzen "DLE DLE ?" (Mode-Abfrage) und "DLE DLE V" (Versions-Abfrage) können im BFZ-STEUER-BASIC nicht verwendet werden, da der Basic-Interpreter die "Antwort" des Interfaces nicht entgegennehmen kann. Sollte der TVI 950-Mode verwendet werden, so kann außerdem die Escape-Sequenz "ESC ?" (Abfrage der Cursor-Position) aus dem gleichen Grund nicht genutzt werden.

Einsatz unter MAT 85

4.3.3 Ein Programmbeispiel in BFZ-STEUER-BASIC

Das nachstehende Programm soll die einzelnen Aus- und Eingabemöglichkeiten des BFZ-STEUER-BASICs in Verbindung mit dem Video-Interface darstellen. Texte werden über eine PRINT-Anweisung ausgegeben, Steuerzeichen mit Hilfe eines Maschinensprache-Programms. Dieses Maschinensprache-Programm wird durch die Programmzeilen 32000 bis 32050 ab Adresse E000 in das RAM geladen. Hier soll nochmals darauf hingewiesen werden, daß dieser Ladevorgang vor dem ersten Aufruf des Maschinensprache-Programms erfolgen muß. Der Aufruf des Maschinensprache-Programms erfolgt über die USR-Anweisung in Programmzeile 32100. Die USR-Anweisung wurde als Unterprogramm geschrieben. Der Code des auszugebenden Steuerzeichens muß vor dem Unterprogramm-Aufruf (also vor GOSUB 32100) in die Variable C geladen werden. Zeile 230 zeigt die Anwendung der "\$"-Funktion.

```
1 GOSUB 32000 : REM *** HILFSROUTINE IN DAS RAM LADEN
10 C=7 : REM *** CODE FUER "BEL"
20 GOSUB 32100 : REM *** CODE AUSGEBEN
30 PRINT "1. TEXTZEILE" : REM *** TEXT AUSGEBEN
40 PRINT : REM EINE LEERZEILE ERZEUGEN
50 PRINT "2. TEXTZEILE"
60 REM *** NUN DEN CURSOR ZWISCHEN DIE BEIDEN TEXTZEILEN STELLEN
70 REM *** (DER CURSOR STEHT NACH DER LETZTEN PRINT-ANWEISUNG
80 REM *** UNTER DEM TEXT "2. TEXTZEILE". DAHER MUSS DER CURSOR
90 REM *** UM ZWEI ZEILEN NACH OBEN GESTELLT WERDEN. DER CODE
100 REM *** FUER "CURSOR HOCH" IST 11 (DEZ))
110 C=11 : GOSUB 32100 : GOSUB 32100
120 REM *** ACHTEN SIE BEIM EINGEBEN DIESES PROGRAMMS DARAUF,
130 REM *** DASS AM ENDE DER NACHFOLGENDEN ZEILE EIN ";" STEHT
140 PRINT "-----";
150 REM *** NUN CURSOR UNTER "2. TEXTZEILE" STELLEN
160 C=10 : REM *** CODE FUER "CURSOR ABWAERTS" (LINE FEED)
170 GOSUB 32100 : GOSUB 32100 : REM *** CURSOR 2 ZEILEN TIEFER
180 C=13 : REM *** CODE FUER "WAGENRUECKLAUF" (CARRIAGE RETURN)
190 GOSUB 32100 : REM *** CURSOR AN ZEILENANFANG
200 INPUT "WIE ALT SIND SIE ? ",A
210 PRINT "SIE BEHAUPTEN, ";A;"JAHRE ALT ZU SEIN"
220 PRINT "BITTE BETAETIGEN SIE EINE TASTE"
230 C=$ : REM *** TASTATURABFRAGE (VERGL. BASIC-ANLEITUNG)
240 PRINT "DIE VON IHNEN BETAETIGTE TASTE ERZEUGT DEN CODE";
250 PRINT C;"(DEZ) BZW. ";"#C;" (HEX)" : REM *** CODE ANZEIGEN
260 PRINT "--- ENDE ---"
31999 END
```

Fortsetzung auf der nächsten Seite

Einsatz unter MAT 85

```
32000 POKE DEC(E000),125
32010 POKE DEC(E001),205
32020 POKE DEC(E002),82
32030 POKE DEC(E003),0
32040 POKE DEC(E004),201
32050 RETURN
32100 X=USR(DEC(E000),C) : RETURN : REM *** ZEICHENAUSGABE
```

Einsatz unter CP/M

5 Einsatz unter dem Betriebssystem CP/M

Zum Betrieb des Video-Interfaces BFZ/MFA 8.4 unter CP/M ist eine Serienschnittstelle erforderlich. Hierzu kann entweder die "V24/RS232-Schnittstelle" BFZ/MFA 4.10 oder die "Programmierbare Serienschnittstelle" BFZ/MFA 4.4 verwendet werden. Bei der "Programmierbaren Serienschnittstelle" sind die V24/TTL- und TTL/V24-Pegelwandler nachzurüsten. Dies ist im FPÜ-Band 4 "CP/M-Ausbaustufe", Kapitel "Aufbau und Inbetriebnahme" näher beschrieben. Eine Serienschnittstelle, die nach der Anleitung im FPÜ-Band 4 für das Video-Interface BFZ/MFA 8.2 eingestellt worden ist, muß noch geringfügig modifiziert werden. Die Modifikation ermöglicht einen Handshake-Betrieb.

- Bei der V24/RS232-Schnittstelle ist eine Drahtbrücke nachzurüsten. Diese Schnittstelle kann nach der Umrüstung sowohl mit dem Video-Interface BFZ/MFA 8.4 als auch mit dem Video-Interface BFZ/MFA 8.2 betrieben werden.
- Bei der Programmierbaren Serienschnittstelle BFZ/MFA 4.4 sind zwei Drahtbrücken nachzurüsten und eine andere zu entfernen. Nach der Umrüstung kann diese Schnittstelle nur noch mit dem hier beschriebenen Video-Interface BFZ/MFA 8.4 betrieben werden (nicht mehr mit dem Video-Interface BFZ/MFA 8.2).

Nachstehend eine kurze Aufstellung der zum Betrieb notwendigen Brücken-Einstellungen und der notwendigen Umrüstung.

Programmierbare Serienschnittstelle BFZ/MFA 4.4	V24/RS232-Schnittstelle BFZ/MFA 4.10
Baudratenbrücke C geschlossen (ergibt 1200 Bd)	Baudratenbr. D geschl. (ergibt 1200 Bd)
geschlossene Lötbrücken: P2 bis P6	geschlossene Lötbrücken: P3, P4, 9, 14, 15, 18
geöffnete Lötbrücken: P1	geöffnete Lötbrücken: 1 bis 8, 10 bis 13, 16, 17
Baugruppe mit Pegelwandlern nachgerüstet (im FPÜ-Band 4, Übung "Aufbau und Betriebnahme" beschrieben)	kein Stecker in die 25pol. Buchsenleiste auf der Frontplatte eingesteckt
Umrüstung:	Umrüstung:
1. <u>Brücke zwischen dem</u> ! <u>Leiterplattenanschluß DSR</u> ! <u>und 0 V (GND) entfernen</u> ! (falls vorhanden)	Drahtbrücke von Anschluß 28c der VG64-Steckerleiste nach Lötbrücke 14
2. Drahtbrücke von Anschluß 28c der VG64-Steckerleiste nach Pin 13 des IC8 (1488)	
3. Drahtbrücke von Pin 11 des IC8 (1488) zum Leiterplatten- anschluß DSR	

Einsatz unter CP/M

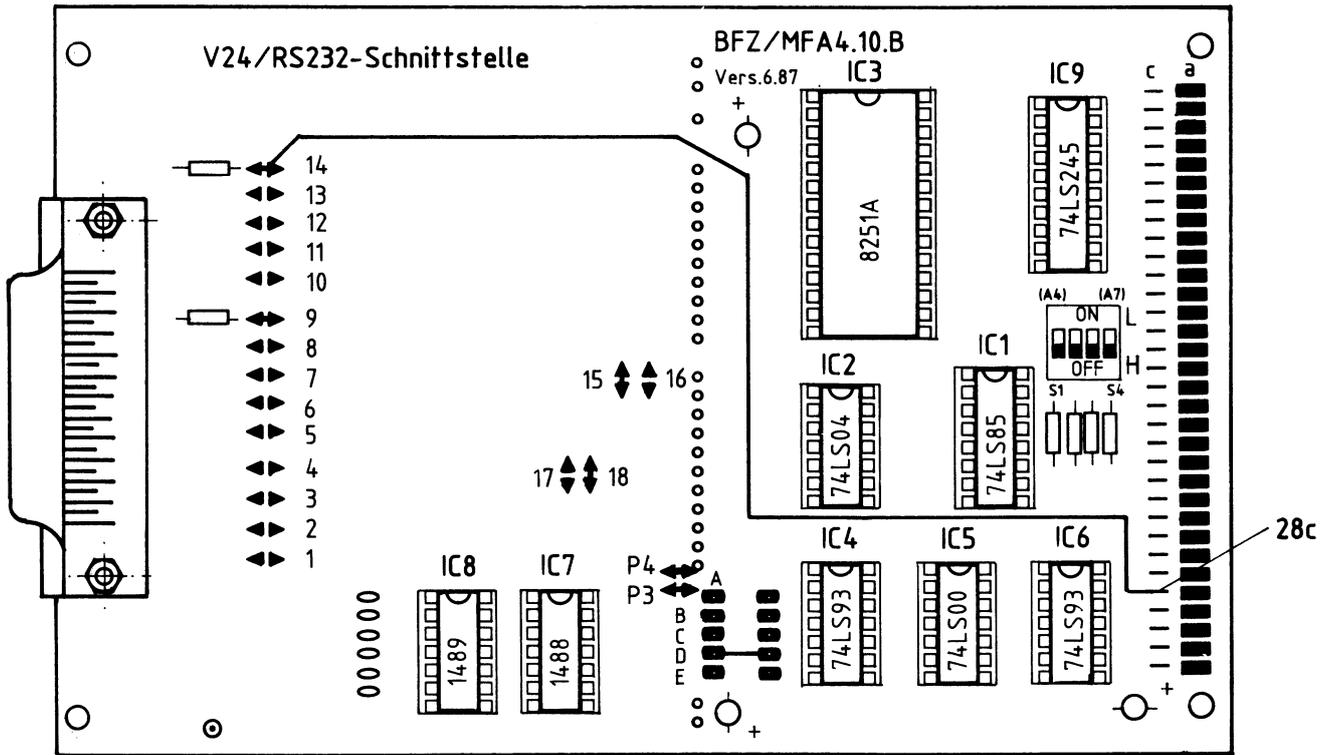


Bild 8: Bestückungsplan der V24/RS232-Schnittstelle

Für den Betrieb der Schnittstelle ist auf der Bestückungsseite der Schnittstelle die eingezeichnete Drahtbrücke nachzurüsten. Sie sollte mit Klebstoff fixiert werden.

Einsatz unter CP/M

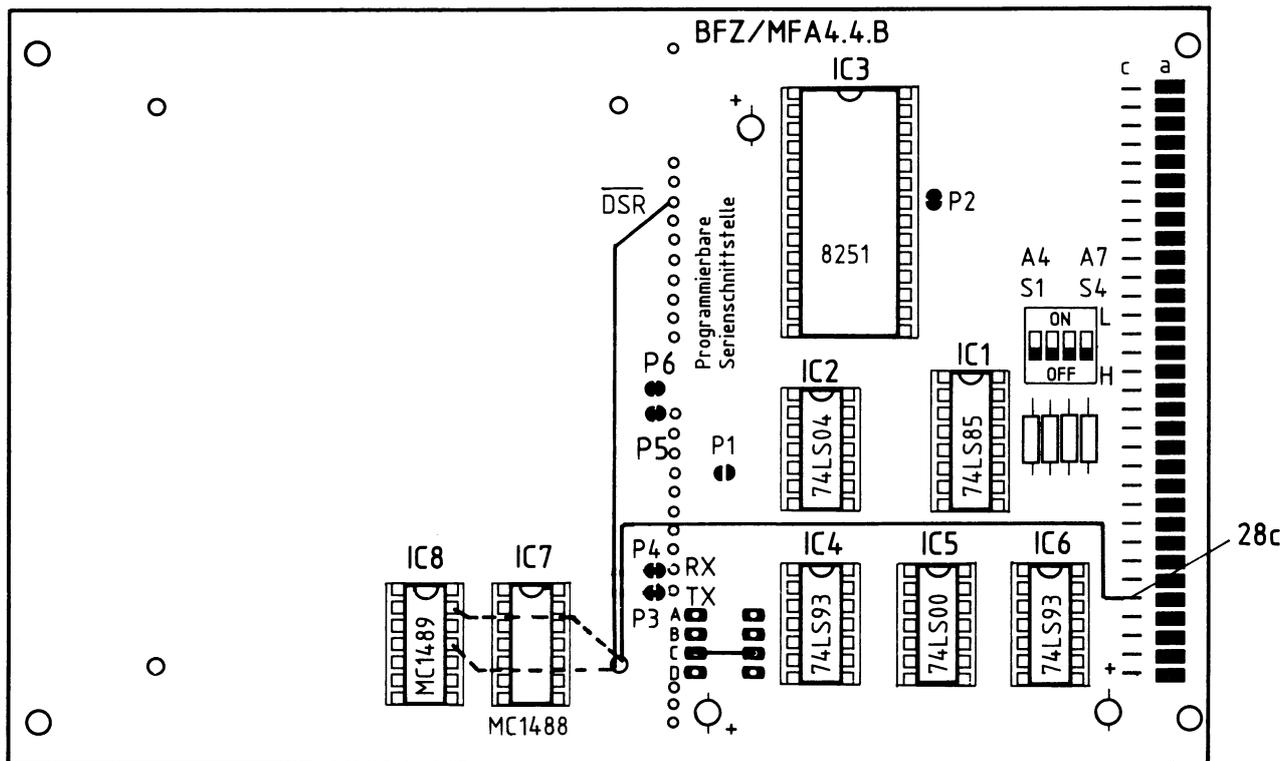


Bild 9: Bestückungsplan der Programmierbaren Serienschnittstelle

Für den Betrieb der Schnittstelle muß die Brücke zwischen dem Leiterplattenanschluß \overline{DSR} und 0 V (GND) entfernt werden! Sie ist deshalb auf dem hier abgebildeten Bestückungsplan nicht eingezeichnet. Stattdessen sind auf der Bestückungsseite der Schnittstelle die zwei eingezeichneten Drahtbrücken nachzurüsten. Die Drahtbrücken sollten mit Klebstoff fixiert werden.

Weitere Hinweise:

Der Datenaustausch zwischen der CPU und dem Video-Interface erfolgt über die zwischengeschaltete Serien-Schnittstelle. Diese ist notwendig, damit das CP/M-Betriebssystem eine Statusabfrage ("Ist ein Zeichen empfangen worden?" bzw. "Kann ein neues Zeichen gesendet werden?") durchführen kann.

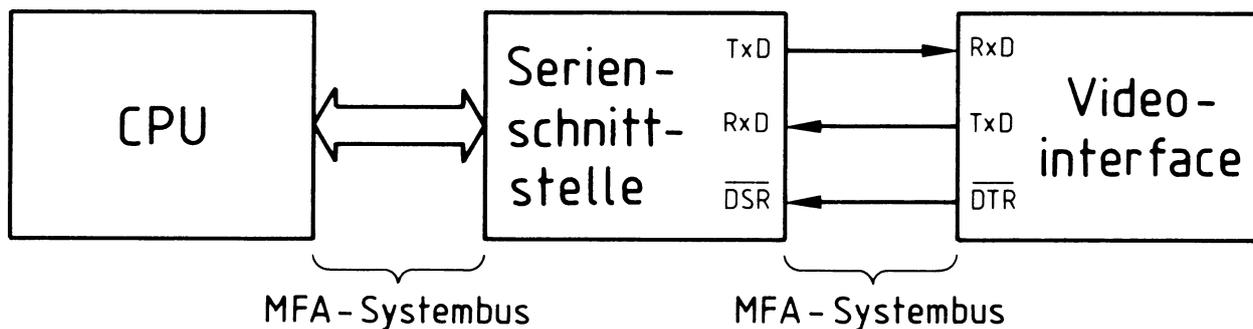


Bild 10: Datenaustausch zwischen CPU und Video-Interface (CP/M)

Einsatz unter CP/M

Vor der Inbetriebnahme unter dem Betriebssystem CP/M sind die Schalter des Video-Interfaces entsprechend dem nachfolgenden Kapitel "Einstellung der Schalter des Video-Interfaces" einzustellen.

Die dort angegebene Schalterstellung gilt für eine Baudrate von 1200 Bd, wie sie entsprechend FPÜ-Band 4 auf den oben erwähnten Serienschnittstellen eingestellt ist. Das hier beschriebene Video-Interface kann aber wahlweise auch mit höheren Baudraten betrieben werden. Mit der "Programmierbaren Serienschnittstelle" (BFZ/MFA 4.4) sind bis zu 4800 Bd, mit der "V24/RS232-Schnittstelle" (BFZ/MFA 4.10) bis zu 9600 Bd möglich. Die Baudratenbrücken der Serienschnittstellen und die DIL-Schalter des Video-Interfaces sind dann entsprechend zu ändern. Informationen über die Einstellung der Baudratenbrücken bzw. der DIL-Schalter können Sie den folgenden Unterlagen entnehmen:

- Für die "Programmierbare Serienschnittstelle" dem FPÜ-Band 2
- Für die "V24/RS232-Schnittstelle" den zur Baugruppe mitgelieferten Unterlagen
- Für das "Video-Interface" dem Anhang dieses Handbuches

Die Kapitel "Nutzung in Assembler Programmen" und "Nutzung unter MBASIC" enthalten Informationen für eigene Assembler- bzw. BASIC-Programme in denen die Steuerfunktionen des Video-Interfaces ausgenutzt werden sollen.

5.1 Einstellung der Schalter des Video-Interfaces

Auf der Interface-Baugruppe befinden sich drei Schalter-Reihen (siehe auch Bestückungsplan im Anhang) zur Grundeinstellung des Video-Interfaces beim Einschalten. Jede der mit S1, S2 und S3 bezeichneten Schalter-Reihen besteht aus acht Schaltern. S1.1 bezeichnet den Schalter Nummer 1 in der Schalter-Reihe S1. Die anderen Schalter sind entsprechend numeriert. Für das Betriebsprogramm MAT 85 sind die Schalter folgendermaßen einzustellen:

- Schalter-Reihe S1:

Schalter	Stellung	eingestellte Funktion
S1.1	OFF	TVI 950-Mode
S1.2	OFF	blinkender Cursor
S1.3	ON	Block-Cursor
S1.4	OFF	dunkler Bildhintergrund
S1.5	OFF	reserviert, immer OFF !
S1.6	OFF	DTR-Handshake
S1.7	OFF	reserviert, immer OFF !
S1.8	OFF	Statuszeile aus

Einsatz unter CP/M

- Schalter-Reihe S2

Schalter	Stellung	eingestellte Funktion
S2.1	ON	Wortlänge 8 Bit
S2.2	OFF	Zeichensatz USA ¹⁾
S2.3	OFF	dto. ¹⁾
S2.4	OFF	dto. ¹⁾
S2.5	ON	Zeilenanzahl 24
S2.6	OFF	dto.
S2.7	ON	Spaltenanzahl 80
S2.8	OFF	dto.

- ¹⁾ Dieser Zeichensatz wird für das Betriebssystem CP/M empfohlen, da hier häufig Zeichen benötigt werden, die im deutschen Zeichensatz nicht enthalten sind. (z.B.: "[" und "]" für Optionen beim PIP-Kommando). Sollten (z.B. für Textverarbeitung) andere Zeichensätze gewünscht werden, so kann der Zeichensatz über das Set-Up-Menü oder über eine Steuersequenz umgeschaltet werden, ohne die Einstellung der Schalter zu verändern. Viele CP/M-Programme (wie z.B. WORDSTAR) können eine solche Umschaltung beim Programmstart automatisch ausführen.

- Schalter-Reihe S3

Schalter	Stellung	eingestellte Funktion
S3.1	ON	2 Stopbits
S3.2	OFF	kein Paritätsbit
S3.3	OFF	dto.
S3.4	OFF	dto.
S3.5	ON	1200 Baud ²⁾
S3.6	OFF	dto. ²⁾
S3.7	ON	dto. ²⁾
S3.8	OFF	dto. ²⁾

- ²⁾ Die hier angegebene Schalterstellung gilt für eine Baudrate von 1200 Bd, wie sie entsprechend FPÜ-Band 4 auf den Serienschnittstellen eingestellt ist. Falls die Serienschnittstellen mit einer anderen Baudrate betrieben werden, ist die Schalterstellung entsprechend zu ändern. Informationen über die Einstellung der DIL-Schalter können Sie dem Anhang dieses Handbuches entnehmen.

Anmerkung:

Die Schalter S1.2 (Cursor-Art), S1.3 (Cursor-Form), S1.4 (Bildhintergrund), S1.8 (Statuszeile) und S2.2, S2.3, S2.4 (alle drei zur Einstellung des Zeichensatzes) können - den persönlichen Wünschen entsprechend - anders eingestellt werden. Eine andere Einstellung dieser Schalter beeinträchtigt die Funktion nicht. Angaben über die durch diese Schalter einstellbaren Optionen entnehmen Sie bitte dem Anhang.

5.2 Nutzung in Assembler-Programmen

5.2.1 Zeichenausgabe

Wenn der TVI 950-Mode unter dem Betriebssystem CP/M verwendet wird, können die BDOS-Funktionen des Betriebssystems zur Zeichenausgabe genutzt werden. Über diese Funktionen (Unterprogramme) können sowohl darstellbare Zeichen (wie Buchstaben und Ziffern) als auch Steuerzeichen ausgegeben werden.

Es gibt verschiedene BDOS-Funktionen zur Zeichenausgabe:

Funktion "CONSOLE OUTPUT", Ausgabe eines einzelnen Zeichens

Anwendung:

- C-Register mit 02_H laden (Funktionscode)
- E-Register mit dem ASCII-Code des auszugebenden Zeichens laden
- BDOS mittels CALL 0005 aufrufen

Funktion "DIRECT CONSOLE I/O", Ausgabe eines einzelnen Zeichens

Anwendung:

- C-Register mit 06_H laden (Funktionscode)
- E-Register mit dem ASCII-Code des auszugebenden Zeichens laden (der Code FF_H ist hier nicht zulässig)
- BDOS mittels CALL 0005 aufrufen

Funktion "PRINT STRING", Ausgabe einer Zeichenfolge

Anwendung:

- C-Register mit 09_H laden (Funktionscode)
- DE-Registerpaar mit der Adresse laden, unter der das erste auszugebende Zeichen im Speicher steht. Die anderen auszugebenden Zeichen müssen in den darauffolgenden Speicherzeilen gespeichert sein. Die Zeichenfolge muß durch das "\$"-Zeichen (Code 24_H, 36_D) abgeschlossen sein. Das "\$"-Symbol selbst wird nicht ausgegeben.
- BDOS mittels CALL 0005 aufrufen

Beispiel:

```
ORG 0100H    ; Anfangsadresse für CP/M-Programme
MVI E,42H   ; Code 42H ("B") in das E-Register laden
MVI C,02H   ; Funktionscode "CONSOLE OUTPUT"
CALL 0005H  ; Zeichen ausgeben
MVI E,0AH   ; Steuer-Code 0AH (Line Feed) laden
MVI C,06H   ; Funktionscode "DIRECT CONSOLE I/O"
CALL 0005H  ; Steuer-Code ausgeben
```

Einsatz unter CP/M

```

        LXI  D,TEXT  ; Zeiger auf Zeichenfolge
        MVI  C,09H   ; Funktionscode "PRINT STRING"
        CALL 0005H   ; Zeichenfolge ausgeben
        JMP  0000H   ; Programm-Ende
;-----
TEXT:   DB   'Dieser Text soll ausgegeben werden'
        DB   '$'    ; Endemarke (wird nicht ausgegeben)

```

Bitte beachten Sie, daß sämtliche Registerinhalte durch das BDOS verändert werden können. Gegebenenfalls sind die Registerinhalte vor dem BDOS-Aufruf (CALL 0005) mit Hilfe von PUSH-Befehlen im Stack zu speichern und nach der Ausführung der BDOS-Funktion durch POP-Anweisungen zu restaurieren.

Genauere Erläuterungen zu den BDOS-Funktionen entnehmen Sie bitte der CP/M-Literatur.

5.2.2 Zeicheneingabe

Alle Zeichen - gleichgültig ob sie über Tastatur eingegeben werden oder die "Antwort" auf eine DLE- bzw. Escape-Sequenz sind - können unter Verwendung von BDOS-Funktionen des Betriebssystems CP/M eingelesen werden.

Hierzu können mehrere BDOS Funktionen benutzt werden:

Funktion "CONSOLE INPUT", Einlesen eines Zeichens mit Echo

Anwendung:

- C-Register mit 01_H laden (Funktionscode)
- BDOS mittels CALL 0005 aufrufen

Diese Funktion wartet, bis ein Zeichen eingegeben wird. Das vom Video-Interface zum Mikrocomputer gesendete Zeichen wird von der BDOS-Funktion sofort zum Interface zurück gesendet und auf dem Bildschirm angezeigt (Echo). Nach der Ausführung der BDOS-Funktion befindet sich das eingelesene Zeichen im Akkumulator der CPU. Diese BDOS-Funktion reagiert in der CP/M-üblichen Weise auf die Eingabe bestimmter Steuerzeichen:

Control S: Consolenausgabe ein/aus

Control P: Drucker ein/aus

Diese Funktion sollte nicht dazu verwendet werden, "Antworten" auf DLE- oder ESC-Sequenzen entgegenzunehmen, da alle vom Mikrocomputer empfangenen Zeichen auf dem Bildschirm angezeigt werden.

Einsatz unter CP/M

Funktion "DIRECT CONSOLE I/O", Einlesen eines Zeichens ohne Echo

Anwendung:

- C-Register mit 06_H laden (Funktionscode)
- E-Register mit FF_H laden (Input)
- BDOS mittels CALL 0005 aufrufen

Diese BDOS-Funktion ist eine kombinierte Ein-/Ausgabe-Funktion. Um die Eingabe zu wählen, muß das E-Register vor dem Funktionsaufruf mit dem Wert FF_H geladen werden. Diese Funktion führt - im Gegensatz zur oben beschriebenen Funktion "CONSOLE INPUT" - kein Echo aus. Das heißt: Das eingegebene Zeichen wird nicht auf dem Bildschirm angezeigt. Darüber hinaus wartet diese Funktion nicht auf eine Tastatur-Betätigung. Wurde keine Taste betätigt, so weist der Akkumulator nach der Rückkehr in das Anwenderprogramm den Inhalt 00_H auf. Aus diesem Grund kann der Code 00_H mit dieser Funktion nicht eingelesen werden. Enthält das Empfangsregister aufgrund einer Tastenbetätigung ein Zeichen, so befindet sich der entsprechende ASCII-Code im Akkumulator.

Diese BDOS-Funktion sollte verwendet werden, wenn die "Antwort" auf eine DLE- oder Escape-Sequenz gelesen werden muß.

Funktion "READ CONSOLE BUFFER", Einlesen einer Zeile in einen Puffer

Anwendung:

- C-Register mit 0A_H laden (Funktionscode)
- DE-Registerpaar mit Adresse des Eingabe-Puffers laden (Pufferlänge = Anzahl der zu puffernden Zeichen plus 2)
- Als erstes Byte in den Puffer die Anzahl der pufferbaren Zeichen eintragen (Pufferlänge-2).
- BDOS mittels CALL 0005 aufrufen

Diese BDOS-Funktion liest eine komplette Eingabezeile in den Puffer ein. Bei der Eingabe können alle üblichen CP/M-Editier-Funktionen genutzt werden (Backspace, Delete, ^C, ^E, ^H, ^J, ^M, ^R, ^U, ^X) Alle eingegebenen Zeichen werden auf dem Bildschirm angezeigt. Durch die Eingabe von "Line Feed" oder "Carriage Return" wird die Eingabe beendet. Das zweite Byte des Puffers enthält dann die Anzahl der tatsächlich eingegebenen Zeichen. Ab dem dritten Byte enthält der Puffer die eingegebenen Zeichen.

Diese Funktion sollte nicht dazu verwendet werden, "Antworten" auf DLE- oder ESC-Sequenzen entgegenzunehmen, da alle vom Mikrocomputer empfangenen Zeichen auf dem Bildschirm angezeigt werden.

Einsatz unter CP/M

Beispiel:

Das folgende Programm zeigt im ersten Teil, wie Zeichen von der Tastatur gelesen werden können. Im zweiten Teil wird eine Abfrage der Software-Version gezeigt.

```

                ORG 0100H    ; Anfangsadresse fuer CP/M-Programme
LOOP:          MVI C,01H    ; Funktionscode "CONSOLE INPUT"
                CALL 0005H  ; Lese Zeichen von der Tastatur in den Akku
                                ; und stelle es auf dem Bildschirm dar
                CPI 0DH     ; Steuer-Zeichen "Carriage Return" eingel.?
                JNZ LOOP    ; Wenn Nein --> LOOP
;-----
; Hier weiter, wenn CR-Taste betaetigt wurde
;
TEIL2:        CALL NEWLIN   ; Gebe CR/LF aus (neue Zeile)
                LXI D,VABFR ; Zeiger auf Steurseq. zur Vers.-Abfrage
                MVI C,09H   ; Funktionscode "PRINT STRING"
                CALL 0005H  ; Steuersequenz ausgeben
                MVI B,05H   ; Laenge der Antwort (5): "V" x "/" y 0DH
                LXI H,BUF   ; Zeiger auf Eingabe-Puffer
                CALL READ   ; Lese Antwort
                MVI M,'$'   ; SchlieÙe Antwort mit '$' ab
                LXI D,BUF   ; Zeiger auf Eingabe-Puffer
                MVI C,09H   ; Funktionscode "PRINT STRING"
                CALL 0005H  ; Zeige "Antwort" an
                JMP 0000H   ; Programm-Ende
;-----
; Unterprogramm, liest Zeichen ("Antwort") in Puffer ein
; Bei Aufruf: HL = Puffer-Adresse
;                B = Anzahl der zu lesenden Zeichen
; Nach UP-Ende: HL = Adr. des letzten gepufferten Zeichens + 1
;                B = 0
;
READ:         PUSH H
                PUSH B
WARTE:        MVI C,06H    ; Funktionscode "DIRECT CONSOLE I/O"
                MVI E,0FFH ; Waehle "Input"
                CALL 0005H  ; Lese Eingabe
                ORA A      ; Wurde ein Zeichen eingegeben ?
                JZ WARTE   ; Nein --> WARTE
                POP B

```

Fortsetzung auf der nächsten Seite

Einsatz unter CP/M

```

        POP  H
        MOV  M,A      ; Schreibe Zeichen in Puffer
        INX  H        ; Stelle Zeiger weiter
        DCR  B        ; Alle Zeichen gelesen ?
        JNZ  READ     ; Nein --> lese naechstes Zeichen
        RET          ; Ja --> Return
;-----
; Unterprogramm, gibt CR/LF aus
;
NEWLIN:  LXI  D,CRLF  ; Zeiger auf Steuerzeichen
        MVI  C,09H   ; Funktionscode "PRINT STRING"
        CALL 0005H   ; Gebe Steuerzeichen aus
        RET
;-----
; Steuerzeichen, Steuer-Sequenzen
;
CRLF:   DB   0DH,0AH  ; Steuerzeichen CR/LF
        DB   '$'      ; Ende-Marke
VABFR:  DB   10H,10H,'V' ; DLE-Sequenz
        DB   '$'      ; Ende-Marke
;-----
; Puffer, 6 Byte
;
BUF:    DB   0,0,0,0,0,0 ; Puffer

```

Bitte beachten Sie, daß sämtliche Registerinhalte durch das BDOS verändert werden können. Gegebenenfalls sind die Registerinhalte vor dem BDOS-Aufruf (CALL 0005) mit Hilfe von PUSH-Befehlen im Stack zu speichern und nach der Ausführung der BDOS-Funktion durch POP-Anweisungen zu restaurieren.

Genauere Erläuterungen zu den BDOS-Funktionen entnehmen Sie bitte der CP/M-Literatur.

5.2.3 Ein Programmbeispiel in Assembler-Sprache

Das nachstehende Programm fordert den Anwender auf, eine Ziffer einzugeben. Der Aufforderungstext wird in dem Unterprogramm "TXTAUS" über die BDOS-Funktion "PRINT STRING" ausgegeben. Die Tastatur wird über die BDOS-Funktion "DIRECT CONSOLE I/O" abgefragt, da kein Echo erwünscht ist. Das Programm überprüft, ob tatsächlich eine Ziffer (ASCII-Code 30_H bis 39_H) eingegeben wurde. Ist dies der Fall, gibt das Programm die entsprechende Ziffer als Wort auf dem Bildschirm aus. Wurde keine Ziffer eingegeben, erscheint die Meldung "KEINE ZIFFER". Das hier beschriebene Programm wertet die Betätigung der "CR"-Taste als Abbruchwunsch aus.

Einsatz unter CP/M

```

ORG 0100H          ;ANFANGSADRESSE FUER CP/M-PROGRAMME
START:  LXI  D,TEXT1      ;ZEIGER AUF 1. TEXT
        CALL TXTAUS      ;TEXT AUSGEBEN
WARTE:  MVI  C,06H        ;FUNKTIONSCODE "DIRECT CONSOLE I/O"
        MVI  E,0FFH      ;WAEHLE "INPUT"
        CALL 0005H      ;LESE EINGABE (OHNE ECHO)
        ORA  A           ;ZEICHEN VON TASTATUR EINGELESEN ?
        JZ   WARTE       ;NEIN --> WARTE
        CPI  0DH         ;CR-TASTE BETAETIGT ?
        JZ   0000H      ;WENN JA --> ENDE
        CPI  30H         ;ASCII-CODE < 30H ?
        JC   NOZIFF      ;WENN JA --> NOZIFF
        CPI  3AH         ;ASCII-CODE >= 3AH ?
        JNC NOZIFF      ;WENN JA --> NOZIFF
        SUI  2FH         ;ASCII-CODE AUF 1...10 (DEZ) REDUZIEREN
        MOV  B,A         ;ERMITTELTEN WERT (ZAEHLER) NACH B
        LXI  D,ZIFF      ;ZEIGER AUF ZIFFERN-TEXTE
SUCHO:  LDAX D           ;ZEICHEN LESEN
        INX  D           ;ZEIGER AUF NAECHSTES ZEICHEN STELLEN
        CPI  '$'         ;'$' (TRENnzeichen) ?
        JNZ  SUCHO       ;WENN NEIN --> SUCHO
        DCR  B           ;ZAEHLER MINUS EINS
        JNZ  SUCHO       ;ZAEHLER NICHT NULL --> SUCHO
        CALL TXTAUS      ;TEXT AUSGEBEN
        JMP  START       ;AUF EIN NEUES
;-----
NOZIFF: LXI  D,NZTXT     ;ZEIGER AUF FEHLERMELDUNG
        CALL TXTAUS      ;MELDUNG AUSGEBEN
        JMP  START       ;AUF EIN NEUES
;-----
TXTAUS: MVI  C,09H       ;FUNKTIONSCODE "PRINT STRING"
        CALL 0005H      ;GEBE ZEICHEN FOLGE AUS
        RET
;-----
TEXT1:  DB   0DH,0AH     ;WAGENRUECKLAUF, ZEILENVORSCHUB
        DB   'BITTE EINE ZIFFER EINGEBEN'
        DB   0DH,0AH     ;WAGENRUECKLAUF, ZEILENVORSCHUB
        DB   '$'        ;ENDEMARKE
NZTXT:  DB   07H         ;BELL
        DB   'KEINE ZIFFER'
        DB   0DH,0AH     ;WAGENRUECKLAUF, ZEILENVORSCHUB
        DB   '$'        ;ENDEMARKE

```

Fortsetzung auf der nächsten Seite

Einsatz unter CP/M

```
ZIFF:   DB   '$'           ;MARKE
         DB   'NULL'
         DB   '$'           ;MARKE
         DB   'EINS'
         DB   '$'           ;MARKE
         DB   'ZWEI'
         DB   '$'           ;MARKE
         DB   'DREI'
         DB   '$'           ;MARKE
         DB   'VIER'
         DB   '$'           ;MARKE
         DB   'FUENF'
         DB   '$'           ;MARKE
         DB   'SECHS'
         DB   '$'           ;MARKE
         DB   'SIEBEN'
         DB   '$'           ;MARKE
         DB   'ACHT'
         DB   '$'           ;MARKE
         DB   'NEUN'
         DB   '$'           ;MARKE
```

5.3 Nutzung unter MBASIC (BASIC 80)

MBASIC ist ein weit verbreiteter Basic-Interpreter für das Betriebssystem CP/M. In diesem Kapitel soll gezeigt werden, wie man die Eigenschaften des Video-Interfaces BFZ/MFA 8.4 in MBASIC-Programmen nutzen kann.

Das Betriebssystem CP/M benötigt nicht unbedingt ein Video-Interface zur Ausgabe von Zeichen, da es auch mit anderen Ausgabe-Geräten (z.B. einem Fernschreiber) zusammenarbeiten kann. Zu den möglichen Ausgabegeräten gehören auch solche, die am Zeilenende nicht automatisch Wagenrücklauf und Zeilenvorschub ausführen. Damit MBASIC auch mit solchen Geräten arbeitet, erzeugt es normalerweise nach jedem 80. ausgegebenen Zeichen die Steuerzeichenkombination Wagenrücklauf/Zeilenvorschub (CR/LF). Hierbei spielt es keine Rolle, ob die zuvor ausgegebenen Zeichen darstellbar waren (Buchstaben, Ziffern usw.) oder ob es sich um Steuerzeichen gehandelt hat. Das Video-Interface benötigt die von MBASIC nach jedem 80. Zeichen erzeugte CR/LF-Kombination nicht. Sie kann sich sogar störend auswirken, weil dadurch nur jede zweite Bildschirmzeile beschrieben wird. Fügt MBASIC die CR/LF-Kombination mitten in eine Steuer-Sequenz ein, so "versteht" das Interface die Sequenz nicht. Aus diesen Gründen muß das automatische Einfügen von CR/LF unterbunden werden. Dies kann durch den am Programmanfang stehenden Befehl

WIDTH 255

erreicht werden.

5.3.1 Zeichenausgabe

Darstellbare Zeichen (wie Buchstaben und Ziffern) können unter MBASIC mit Hilfe der PRINT-Anweisung ausgegeben werden:

```
10 PRINT "Video-Interface"
```

Die Ausgabe von Steuerzeichen ist durch die PRINT- und CHR\$-Anweisungen möglich:

```
10 STEUERCODE=7 : PRINT CHR$(STEUERCODE) ;  
20 PRINT CHR$(7) ;
```

Die oben abgedruckten Zeilen sind gleichwertig, sie geben alle das Steuerzeichen BEL (ASCII-CODE 7_D) aus. Die PRINT-Anweisungen sind jeweils mit einem ";" abgeschlossen, damit MBASIC nach dem auszugebenden Steuerzeichen BEL nicht zusätzlich die Steuerzeichen für Wagenrücklauf und Zeilenvorschub ausgibt.

Beachten Sie bitte die CHR\$-Anweisung! Ohne diese Anweisung würde die Ziffer "7" ausgegeben:

```
10 PRINT 7; : REM *** Diese Anweisung gibt die Ziffer "7" aus
```

Einsatz unter CP/M

Sollen Steuerzeichen-Folgen (z.B. Escape-Sequenzen) ausgegeben werden, so kann hierzu eine einzige PRINT-Anweisung verwendet werden. Die einzelnen Steuerzeichen müssen - soweit sie nicht in einem *String* zusammengefaßt sind - durch ein ";" (**nicht durch ein " , " !**) voneinander getrennt werden:

```
10 PRINT CHR$(27);CHR$(71);CHR$(52);
```

Das Semikolon zwischen den einzelnen Zeichen verhindert, daß MBASIC Leerzeichen in die zum Video-Interface übertragene Zeichenfolge einfügt.

Die oben angegebene Zeichenfolge schaltet die inverse Zeichendarstellung ein. Auch andere Möglichkeiten zur Ausgabe von Steuerzeichenfolgen sind erlaubt:

```
10 PRINT CHR$(27);"G4";
```

oder

```
10 ESC$=CHR$(&H1B)
20 PRINT ESC$;"G4";
```

oder

```
10 INVEIN$=CHR$(27)+"G4"
20 PRINT INVEIN$;
```

usw.

5.3.2 Zeicheneingabe

Um Tastatur-Eingaben in MBASIC entgegenzunehmen, können die Basic-Anweisungen INPUT, INPUT\$ und INKEY\$ verwendet werden. Eine genaue Beschreibung dieser Anweisungen entnehmen Sie bitte der MBASIC-Anleitung.

Beispiele:

```
10 INPUT "Eingabe",I1$
20 I2$=INPUT$(4) : REM *** LIEST 4 ZEICHEN
30 I3$=INKEY$ : IF I3$="" THEN 30
```

"Antworten" auf DLE- bzw. Escape-Sequenzen sollten **ausschließlich mit der INPUT\$-Anweisung** entgegengenommen werden. Dabei sollte - wie das nachfolgende Beispiel zeigt - die INPUT\$-Anweisung direkt auf die PRINT-Anweisung zur Ausgabe der DLE- bzw. Escape-Sequenz folgen. Bitte achten Sie darauf, daß in der Klammer der INPUT\$-Anweisung die richtige Anzahl der "Antwort"-Zeichen angegeben wird.

Einsatz unter CP/M

```
1 WIDTH 255 : REM *** UNTERDRUECKE VOM INTERPRETER-CR/LF
10 ESC$=CHR$(&H1B) : REM *** ESCAPE
20 WO$=ESC$+"?" : REM *** ESC-SEQUENZ ZUR CURSOR-POSITIONS-ABFR.
30 REM *** FUER DIESES BEISPIEL IST ES WICHTIG, DASS DIE NACH-
40 REM *** FOLGENDE PRINT-ANWEISUNG MIT EINEM ";" ABGESCHLOSSEN
50 REM *** IST.
60 PRINT "Bitte CONTROL-C betaetigen"; : REM *** Text ausgeben
70 PRINT WO$; : REM *** ABFRAGE DER CURSOR-POSITION (mit ";" !!)
80 I$=INPUT$(3) : REM *** "ANTWORT" ENTGEGENNEHMEN (3 ZEICHEN)
90 REM *** DIE ANTWORT BESTEHT AUS 3 ZEICHEN:
100 REM *** REIHE+20H, SPALTE+20H, ODH
110 REIHE=ASC(LEFT$(I$,1))-&H20
120 SPALTE=ASC(MID$(I$,2,1))-&H20
130 REM *** NUN DIE CURSOR-POSITION ANZEIGEN
140 PRINT "Reihe=";REIHE;", Spalte=";SPALTE
150 REM *** NUN CURSOR AUF DIE ALTE POSITION, DIE ER VOR DER
160 REM *** ANZEIGE VON "REIHE" UND "SPALTE" HATTE
170 PRINT ESC$;"=";LEFT$(I$,2);
180 REM *** DIE ERSTEN BEIDEN ZEICHEN VON I$ GEBEN DIE POSITION AN
190 REM *** ZUM ABSCHLUSS EINE ENDLOS-SCHLEIFE, DAMIT DIE
200 REM *** CURSOR-POSITION NICHT DURCH DIE READY-MELDUNG DES
210 REM *** BASIC-INTERPRETERS VERAENDERT WIRD (PROGRAMM BITTE
220 REM *** MIT "CONTROL-C" ABBRECHEN)
230 GOTO 230
```

5.3.3 Ein Programmbeispiel in MBASIC (BASIC 80)

Das nachstehende Programm soll die einzelnen Aus- und Eingabemöglichkeiten unter MBASIC darstellen. Texte und Steuerzeichen werden mit Hilfe einer PRINT-Anweisung ausgegeben. Zeile 190 zeigt die Anwendung der INKEY\$-Funktion.

```
1 WIDTH 255 : REM *** UNTERDRUECKE INTERPRETER-CR/LF
10 PRINT CHR$(7); : REM *** CODE FUER "BEL" AUSGEBEN
20 PRINT "Dies ist ein Test" : REM *** TEXT AUSGEBEN
30 PRINT : REM EINE LEERZEILE ERZEUGEN
40 PRINT "Gross- und Kleinbuchstaben"
50 REM *** NUN DEN CURSOR ZWISCHEN DIE BEIDEN TEXTZEILEN STELLEN
```

Fortsetzung auf der nächsten Seite

Einsatz unter CP/M

```
60 REM *** (DER CURSOR STEHT NACH DER LETZTEN PRINT-ANWEISUNG
70 REM *** UNTER DEM TEXT "Gross - ....". DAHER MUSS DER CURSOR
80 REM *** UM ZWEI ZEILEN NACH OBEN GESTELLT WERDEN. DER CODE
90 REM *** FUER "CURSOR HOCH" IST 11 (DEZ))
100 PRINT CHR$(11);CHR$(11);
110 REM *** ACHTEN SIE BEIM EINGEBEN DIESES PROGRAMMS DARAUFG,
120 REM *** DASS AM ENDE DER NACHFOLGENDEN ZEILE EIN ";" STEHT
130 PRINT "-----";
140 REM *** NUN CURSOR UNTER "Gross- und Kleinbuchstaben"
150 PRINT CHR$(10);CHR$(10);CHR$(13);: REM *** LF, LF, CR
160 INPUT "WIE ALT SIND SIE ",A
170 PRINT "SIE BEHAUPTEN, ";A;"JAHRE ALT ZU SEIN"
180 PRINT "BITTE BETAETIGEN SIE EINE TASTE"
190 C$=INKEY$ : IF C$="" THEN 190 : REM *** TASTATURABFRAGE
200 PRINT "DIE VON IHNEN BETAETIGTE TASTE ERZEUGT DEN CODE";
210 PRINT ASC(C$);" (DEZ)" : REM *** CODE ANZEIGEN
220 PRINT "--- ENDE ---"
```

5.4 Das Editor-Programm BFZED.COM

Der zum CP/M-Betriebssystem für den MFA-Mikrocomputer mitgelieferte Editor "BFZED.COM" ist für das Video-Interface BFZ/MFA 8.2 installiert. Dies bedeutet, daß zur Steuerung des Cursors der MAT 85-Mode des hier beschriebenen Interfaces benutzt werden muß. Dieser Mode kann vor dem Start des Editors über das Set-Up-Menü ein- und nach Verlassen des Editors wieder ausgeschaltet werden. Abgesehen davon, daß hierbei ständig zwischen den beiden Modi hin- und hergeschaltet werden muß, können die erweiterten Möglichkeiten des TVI 950-Modus so nicht genutzt werden.

Deshalb ist es sinnvoll, das Programm BFZED.COM an den TVI 950-Mode anzupassen. Bevor man diese Anpassung jedoch durchführt, sollte man sich eine Sicherheitskopie des Programms anfertigen.

Arbeiten Sie NIE mit der Original-Diskette (außer zum Erstellen einer Sicherheitskopie)

Bedenken Sie ebenfalls, daß die geänderte BFZED.COM-Version ausschließlich im TVI 950-Mode des hier beschriebenen Video-Interfaces BFZ/MFA 8.4 funktionsfähig ist. Sollten Sie BFZED.COM auch auf dem Video-Interface BFZ/MFA 8.2 betreiben wollen, so müssen Sie hierfür die ungeänderte Version des Programms benutzen.

Einsatz unter CP/M

5.4.1 Vorbereitungen zum Anpassen des Programms BFZED.COM

- Kopieren Sie folgende Programme auf eine formatierte und mit dem CP/M-Betriebssystem versehene Diskette: STAT.COM, DDT.COM, BFZED.COM und BFZED.HLP.
Alle Arbeiten sollten nur mit dieser neu geschaffenen Arbeitsdiskette durchgeführt werden, damit nicht versehentlich die Original-Version des Editors verändert wird.
- Setzen Sie die neue Arbeitsdiskette in Laufwerk A: ein und betätigen Sie die RESET-Taste des Mikrocomputers. Nach wenigen Sekunden muß die CP/M-Bereitschaftsmeldung "A>" erscheinen.
- Ermitteln Sie die Länge des Programms BFZED.COM mit Hilfe des Programms STAT.COM. Geben Sie hierzu die Anweisung

```
STAT BFZED.COM
```

ein und schließen Sie die Eingabe durch die Betätigung der CR-Taste ab.
Auf dem Bildschirm wird daraufhin beispielsweise

```
RECS    BYTES    EXT    ACC
      64      8K      1    R/W A:BFZED.COM
BYTES REMAINING ON A: 288K
```

angezeigt. Hiervon ist nur der Wert unter "RECS" wichtig. Er gibt die Länge des Programms BFZED.COM in Records zu je 128 Bytes an. Falls der angezeigte Wert ungerade ist, muß er um 1 erhöht werden. Dividieren Sie den (ggf. um 1 erhöhten) Wert durch 2 und notieren Sie das Divisions-Ergebnis.

Beispiele:

Angezeigter Wert	plus 1 (da ungerade)?	Divisions-Ergebnis (Notieren !)
63	JA --> 64	32
64	NEIN	32

5.4.2 Anpassen des Programms BFZED.COM

- Geben Sie nach den oben angegebenen Vorbereitungen nun die folgende Kommandozeile ein:

```
DDT BFZED.COM
```

Schließen Sie die Eingabe mit der CR-Taste ab. Die Eingabe bewirkt, daß zuerst das Monitorprogramm DDT.COM geladen wird. Dieses lädt dann das Editor-Programm BFZED.COM in den RAM-Speicher. Dort kann es verändert werden.

Nachstehend ist der "Dialog" zwischen Benutzer und DDT.COM abgedruckt. Ausgaben des Programms DDT.COM sind in der Schrift *Courier* gesetzt. Eingaben des Benutzers sind in der

Einsatz unter CP/M

unterstrichenen Schrift Courier gesetzt. Alle Benutzereingaben sind durch die Betätigung der CR-Taste abzuschließen. Das Zeichen "¿" im folgenden Bildschirm-Dialog symbolisiert DDT-Ausgaben, die - abhängig von der jeweiligen BFZED.COM-Version - unterschiedlich sein können. Kommentare sind in der Schrift *Times Roman Italic* gesetzt.

```
DDT VERS ¿.¿  Meldung des Programms DDT.COM
NEXT  PC
¿¿¿¿ ¿¿¿¿
-S0106      Benutzereingabe mit CR-Taste abschließen !
0106 ¿¿ 4D  Anzahl der Zeichen pro Zeile (hex.)
0107 ¿¿ 17  Anzahl der Zeilen pro Bildschirmseite (hex.)
0108 ¿¿ 0   Keine Zeitverzögerung nach dem Löschen d. Bildschirms
0109 ¿¿ .   Punkt "." eingeben, anschließend auch hier CR betätigen
-S012F
012F ¿¿ 1   Ein Steuerbyte für Cursor nach rechts
0130 ¿¿ 0C  Steuerbyte: ^L
0131 ¿¿ .   Punkt "." eingeben, anschließend auch hier CR betätigen
-S013E
013E ¿¿ 1   Ein Steuerbyte für Bildschirm löschen, Home Cursor
013F ¿¿ 1A  Steuerbyte: ^Z
0140 ¿¿ .   Punkt "." eingeben, anschließend auch hier CR betätigen
-S0143
0143 ¿¿ 3   Drei Steuerbyte für Carriage Return mit Löschen des Zeilenrestes
0144 ¿¿ 1B  1. Steuerbyte: Escape
0145 ¿¿ 54  2. Steuerbyte: ASCII-Code für "T"
0146 ¿¿ 0D  3. Steuerbyte: Wagenrücklauf
0147 ¿¿ .   Punkt "." eingeben, anschließend auch hier CR betätigen
-S0148
0148 ¿¿ 0D  Steuerzeichen für Carriage Return ohne Löschen des Zeilenrestes
0149 ¿¿ .   Punkt "." eingeben, anschließend auch hier CR betätigen
-
```

- Nachdem alle Änderungen durchgeführt wurden, kann das Programm DDT.COM durch die Tastenkombination CONTROL-C beendet werden.
- Nun ist es wichtig, die noch im RAM befindliche neue Version des Programms BFZED.COM auf der Diskette zu speichern. erinnern Sie sich noch an den Wert, den Sie sich notieren sollten? Diesen Wert brauchen wir jetzt! Geben Sie die Kommando-Zeile

SAVE ¿¿ BFZED950.COM

ein ("¿¿" symbolisiert den notierten Wert). Das Betriebssystem CP/M "weiß" nun, wie groß der RAM-Bereich ist, der auf der Diskette gespeichert werden muß. Die neue Version des Editor-Programms wird unter dem Namen BFZED950.COM gespeichert.

- Anschließend kann die neue Version des Editors aufgerufen und überprüft werden.

Einsatz unter CP/M

- Die hier durchgeführten Änderungen passen den Editor lediglich an das neue Bildformat (80 Zeichen pro Zeile, 24 Zeilen) an. Soll der Editor z.B. beim Programmaufruf automatisch vom USA-Zeichensatz zum deutschen Zeichensatz umschalten, so ist auch dies möglich. Die Steuerzeichen für die zusätzlichen Anpassungen entnehmen Sie bitte diesem Handbuch. Der Änderungsvorgang selbst ist im FPÜ-Band 4 ("Beschreibung des Programms BFZED.COM") erläutert.

Die Statuszeile

6 Die Statuszeile

Das hier beschriebene Interface kann zusätzlich zu den Zeichen-Zeilen am unteren Rand des Schriftfeldes eine Statuszeile anzeigen. Diese kann mit dem Schalter S1.8 oder mit Hilfe des Set-Up-Menüs ein- bzw. ausgeschaltet werden. Die Statuszeile ist unabhängig vom eingeschalteten Mode (MAT 85 bzw. TVI 950). Sie steht dem Anwender nicht zur Zeichenausgabe zur Verfügung, das heißt, er kann in diese zusätzliche Zeile keinen Text schreiben. In der Statuszeile wird mit halber Helligkeit und dunkler Schrift auf hellem Grund eine Zustandsinformation des Video-Interfaces angezeigt.

Die Statuszeile besteht aus acht Spalten und hat folgenden Aufbau:

Control-Mode	Parität	Datenbits	Stopbits	Baudrate	Zeichensatz	Emulations-Mode	Rrr Css
CTL-MODE	P:OFF	D:8	S:2	1200 BAUD	USA	TVI950	R02 C17

Die Überschriften der einzelnen Spalten werden auf dem Bildschirm nicht angezeigt. Die hier dargestellte Anzeige ist ein Beispiel. Die tatsächlich angezeigten Werte sind abhängig von der Einstellung des Video-Interfaces bzw. von der Cursor-Position.

Control-Mode:

Wenn der Control-Mode aktiviert wurde, zeigt das Interface in dieser Spalte die Meldung "CTL-MODE" an. Normalerweise ist dieses Feld leer.

Parität:

Hinter den Zeichen "P:" wird der aktuelle Paritäts-Mode angezeigt. Möglich sind die Anzeigen "OFF" (keine Parität), "ODD" (odd Parity, ungerade Parität), "EVN" (even Parity, gerade Parität), "MRK" (Mark, immer H-Pegel), "SPC" (Space-Parity, immer L-Pegel). Der Paritäts-Mode kann über die DIL-Schalter der Baugruppe eingestellt werden.

Datenbits:

In dieser Spalte wird hinter den Zeichen "D:" die Anzahl der Datenbits pro übertragenem Zeichen angezeigt. Möglich sind "7" bzw. "8" Datenbits. Die Anzahl der Datenbits kann über die DIL-Schalter der Baugruppe eingestellt werden.

Stopbits:

Die eingestellte Anzahl der Stopbits ("1" oder "2") wird hinter den Zeichen "S:" angezeigt. Die Anzahl der Stopbits kann über die DIL-Schalter der Baugruppe eingestellt werden.

Baudrate:

Die eingestellte Baudrate wird in dieser Spalte angezeigt. Folgende Baud-Raten sind möglich: 110 Bd; 134,5 Bd; 150 Bd; 300 Bd; 600 Bd; 1200 Bd; 1800 Bd; 2400 Bd; 3600 Bd; 4800 Bd; 7200 Bd; 9600 Bd; 19200 Bd.

Die Statuszeile

Zeichensatz:

Der aktuelle länderspezifische Zeichensatz wird hier durch eine entsprechende Abkürzung angezeigt:

<u>USA</u>	Vereinigte Staaten von Amerika
<u>FR</u> ance	Frankreich
<u>GER</u> many	Deutschland
<u>EN</u> gland	England
<u>DEN</u> mark	Dänemark
<u>SW</u> eden	Schweden
<u>IT</u> aly	Italien
<u>SP</u> ain	Spanien

Emulations-Mode:

In dieser Spalte wird der aktuelle Emulations-Mode ("MAT85" oder "TVI950") angezeigt.

Rrr Css (Cursor-Position):

Im rechten Teil der Status-Zeile wird die aktuelle Cursor-Position angezeigt. Hinter "R" (für "ROW", Reihe) wird die Nummer der Reihe, in der sich der Cursor momentan befindet, angezeigt. Hinter "C" (für "COLUMN", Spalte) erfolgt die Anzeige der Spalten-Nummer. Die für diese Anzeige gültige Reihen- und Spaltenzählung beginnt bei 1.

Hinweis:

Wie beim original TVI 950-Terminal, beginnt die Spalten- und Reihenzählung zur Abfrage der Cursor-Position (ESC ?) und zur Positionierung des Cursors (ESC =) bei 0, die Zählung für die Cursor-Positionsanzeige in der Status-Zeile aber bei 1.

 Das Set-Up-Menü

7 Das Set-Up-Menü

Viele Einstellungen, die über DIL-Schalter bzw. Escape-Sequenzen möglich sind, lassen sich auch über die Tastatur verändern. Hierzu muß das *Set-Up-Menü* (Einstell-Menü) durch die Eingabe von CONTROL-DEL aufgerufen werden (CONTROL-Taste gedrückt halten, zusätzlich die im rechten Teil der Tastatur befindliche DEL-Taste betätigen (nicht die DEL-Taste links oben auf der Tastatur), anschließend beide Tasten loslassen).

Nach dem Aufruf des Set-Up-Menüs erscheint folgende Anzeige auf dem Bildschirm:

<u>SETUP Video-Interface BFZ/MFA 8.4 Vx.x</u>			
Auswahl mit "Auf"/"Ab", ändern mit Leertaste, Ende mit CONTROL <u>DEL</u>			
<u>TERMINAL</u>		<u>VIDEO</u>	
Emulation	TVI950	Hintergrund	DUNKEL
Spalten	80	Attr. Bit 7	AUS
Zeilen	24	<u>CURSOR</u>	
Zeichensatz	USA	Form	BLOCK
Statuszeile	AUS	Art	BLINK
Control Mode	AUS		

Die oben dargestellte Bildschirm-Anzeige zeigt eine mögliche Einstellung des Video-Interfaces. Je nach der Einstellung des von Ihnen verwendeten Interfaces kann die Anzeige an einigen Stellen davon abweichen.

Die Anzeige soll nun im einzelnen erläutert werden:

SETUP Video-Interface BFZ/MFA 8.4 Vx.x

Die erste Zeile ist die Meldung des Set-Up-Menüs. In ihr wird auch die Version des auf dem Interface verwendeten EPROMs angezeigt (hier mit "vx.x" angedeutet).

Die zweite Zeile gibt einige Bedienungshinweise:

Auswahl mit "Auf"/"Ab", ändern mit Leertaste, Ende mit CONTROL DEL

 Das Set-Up-Menü

Der restliche Teil des Set-Up-Menüs ist in zwei Spalten aufgeteilt. In der ersten Zeile der linken Spalte befindet sich der Text

```

                Emulation      TVI950
  
```

Der Textteil hinter dem Wort "Emulation" (entweder - wie hier - TVI950 oder MAT85) ist invers dargestellt. Dieser inverse Block kann über die Pfeiltasten "Auf" und "Ab" der Tastatur in die anderen Zeilen der Spalte bewegt werden (versuchen Sie es einmal!). Befindet sich der inverse Block in der untersten Zeile einer Spalte, springt er bei Betätigung der "Ab"-Taste in die oberste Zeile der Nachbar-Spalte. Betätigt man nun die "Auf"-Taste, springt er zurück in die alte Position. Durch das Positionieren des inversen Blocks kann man auswählen, welcher Parameter (Emulation, Spalten, Zeilen ...) geändert werden soll.

Eine Änderung der einzelnen Parameter wird durch Betätigung der Leertaste erreicht. Bei jeder Betätigung wird eine neue Einstellung innerhalb des inversen Blocks angezeigt. Die Leertaste muß so oft betätigt werden, bis die gewünschte Einstellung erscheint. Sind auf diese Weise alle möglichen Einstellungen eines Parameters der Reihe nach zur Anzeige gebracht worden, erscheint wieder die erste Einstellung. Die eingestellten Parameter werden erst beim Verlassen des Set-Up-Menüs aktiv.

Wenn alle Parameter entsprechend den persönlichen Wünschen eingestellt sind, kann das Set-Up-Menü durch erneute Betätigung von CONTROL-DEL verlassen werden. Dabei wird der alte Bildschirminhalt - soweit nach der Parameteränderung möglich - restauriert, und die Zeichendarstellung den aktuellen Parametern angepaßt. Eine Änderung des länderspezifischen Zeichensatzes wirkt sich erst bei der Übertragung weiterer Zeichen aus.

Folgende Einstellungen sind möglich:

```

Emulation      : TVI950 / MAT85
Spalten        : 72 / 80 / 88 / 96
Zeilen         : 22 / 24 / 26 / 28
Zeichensatz    : USA / FRA / GER / ENG / DEN / SWE / ITA / SPA
Statuszeile    : AUS / EIN
Control Mode   : AUS / EIN
Hintergrund    : HELL / DUNKEL
Attribut Bit 7 : AUS / REVERS / BLINK / UNDERL
Cursor-Form    : STRICH / BLOCK
Cursor-Art     : BLINK / RUHEND / AUS
  
```

Das Set-Up-Menü

Die einzelnen Einstellmöglichkeiten sprechen größtenteils für sich oder sind bereits an anderer Stelle erläutert worden. Deshalb soll hier lediglich auf die Parameter "Hintergrund" und "Attribut Bit 7" näher eingegangen werden. Der Control-Mode wird im folgenden Kapitel erläutert.

Über "Hintergrund" kann der Bildhintergrund eingestellt werden. Bei "Hintergrund DUNKEL" werden helle Zeichen auf dunklem Grund und bei "Hintergrund HELL" dunkle Zeichen auf hellem Grund dargestellt.

Zur Übertragung von ASCII-Zeichen werden nur sieben Bit benötigt (Bit 0...6). Über das Bit 7 kann dem übertragenem Zeichen ein Attribut zugeordnet werden. Mit Hilfe des Parameters "Attribut Bit 7" wird festgelegt, um welches Attribut es sich dabei handelt. Ist z.B. "Attribut Bit 7 UNDERL" gewählt, werden alle Zeichen, deren Bit 7 = "1" ist, unterstrichen dargestellt.

Beispiel:

Übertragenes Zeichen	angezeigtes Zeichen
01000001 _B = 41 _H	A
11000001 _B = C1 _H	<u>A</u>

Bei der Einstellung "Attribut Bit 7 AUS" wird das Bit 7 vom Video-Interface ignoriert. Die Möglichkeit, dem Bit 7 ein bestimmtes Attribut zuzuordnen, kann unter dem Betriebsprogramm MAT 85 nicht genutzt werden. Hier werden bei der Zeichenübertragung nur die Bits 0 bis 6 genutzt.

Hinweis:

Bei der Tastatureingabe CONTROL-DEL wird der Code 1F_H erzeugt, der das Set-Up-Menü einschaltet. Der Code 1F_H selbst wird hierbei nicht zum Mikrocomputer übertragen. Soll dieser Code trotzdem zum Mikrocomputer gesendet werden, ist folgendermaßen vorzugehen:

- Set-Up-Menü durch CONTROL-DEL einschalten

Das Set-Up-Menü wird angezeigt

- CONTROL-@ eingeben

Der Code 1F_H wird zum Mikrocomputer gesendet, das Set-Up-Menü wird abgeschaltet

Der Control-Mode

8 Der Control-Mode

Im Control-Mode werden alle **Steuerzeichen**, die das Video-Interface empfängt, nicht ausgeführt, sondern als inverse Zeichen mit halber Helligkeit auf dem Bildschirm angezeigt. Darstellbare Zeichen werden unverändert angezeigt. Mit Hilfe dieses Modes kann der Anwender feststellen, welche Steuerzeichen vom Mikrocomputer zum Video-Interface gesendet werden. So ist eine einfache Überprüfung der Anwender-Software möglich.

Die folgende Tabelle zeigt einige häufig benutzte Steuerzeichen und ihre Darstellung (eine vollständige Tabelle finden Sie im Anhang):

Code	Mnemonic	angez. Zeichen ¹⁾
07 _H	BEL	G
08 _H	BS	H
09 _H	HT	I
0A _H	LF	J
0B _H	VT	K
0C _H	FF	L
0D _H	CR	M
10 _H	DLE	P
16 _H	SYN	V
1A _H	SUB	Z
1B _H	ESC	[
1E _H	RS	^

¹⁾ Die angezeigten Zeichen sind abhängig vom eingestellten länderspezifischen Zeichensatz. Für diese Tabelle wurde der USA-Zeichensatz (empfohlene Grundeinstellung und Tastaturbeschriftung) verwendet.

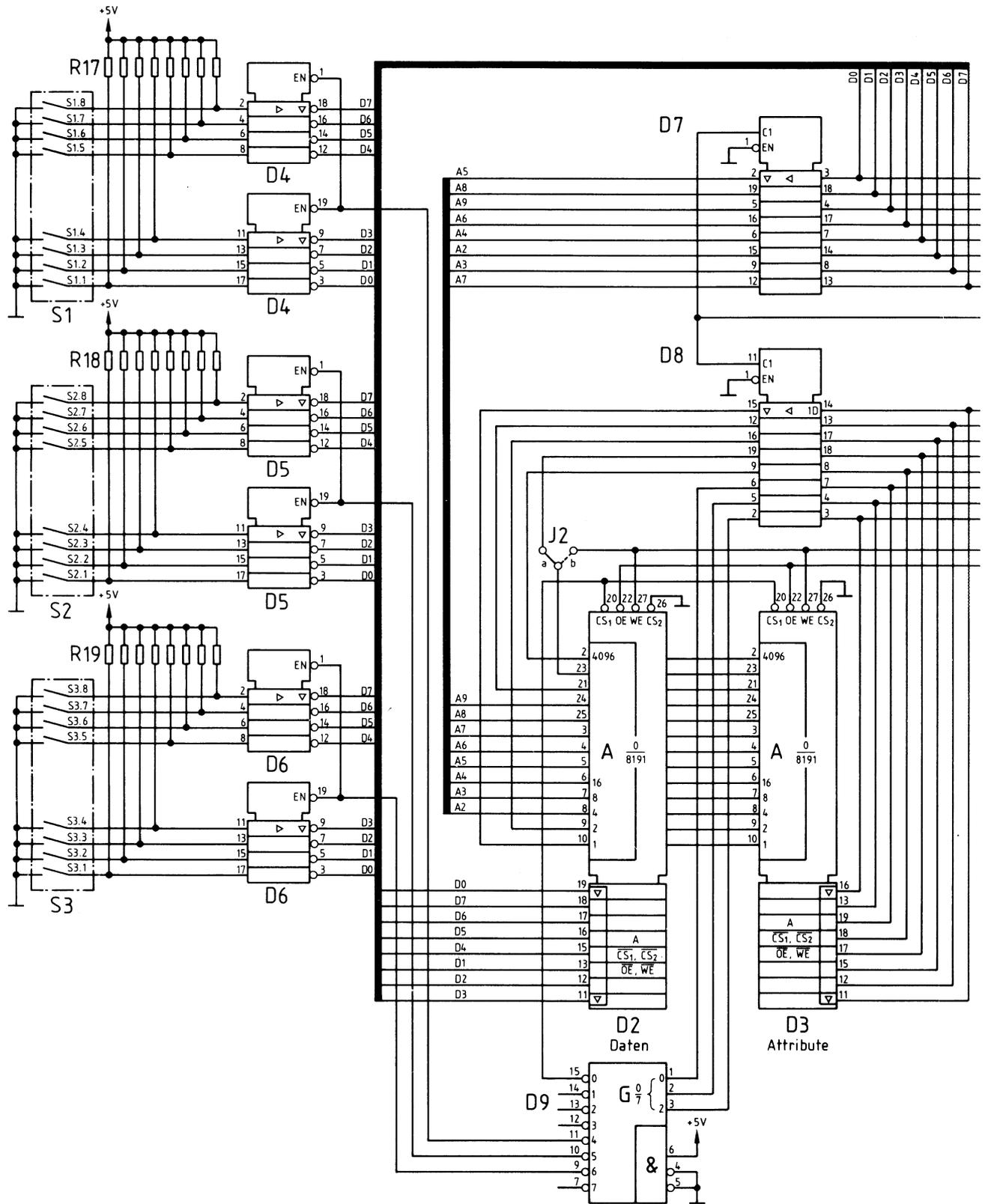
Der Control-Mode kann auf zwei verschiedene Arten aktiviert werden:

- durch Einschalten von "Control Mode EIN" im Set-Up-Menü
- durch Senden der Escape-Sequenz "ESC U" im TVI 950-Mode

Hinweis:

Da im Control-Mode alle Steuerzeichen nur noch angezeigt aber nicht ausgeführt werden, kann der Control-Mode nur über das Set-Up-Menü (Control Mode AUS) verlassen werden. Siehe auch: "Das Set-Up-Menü".

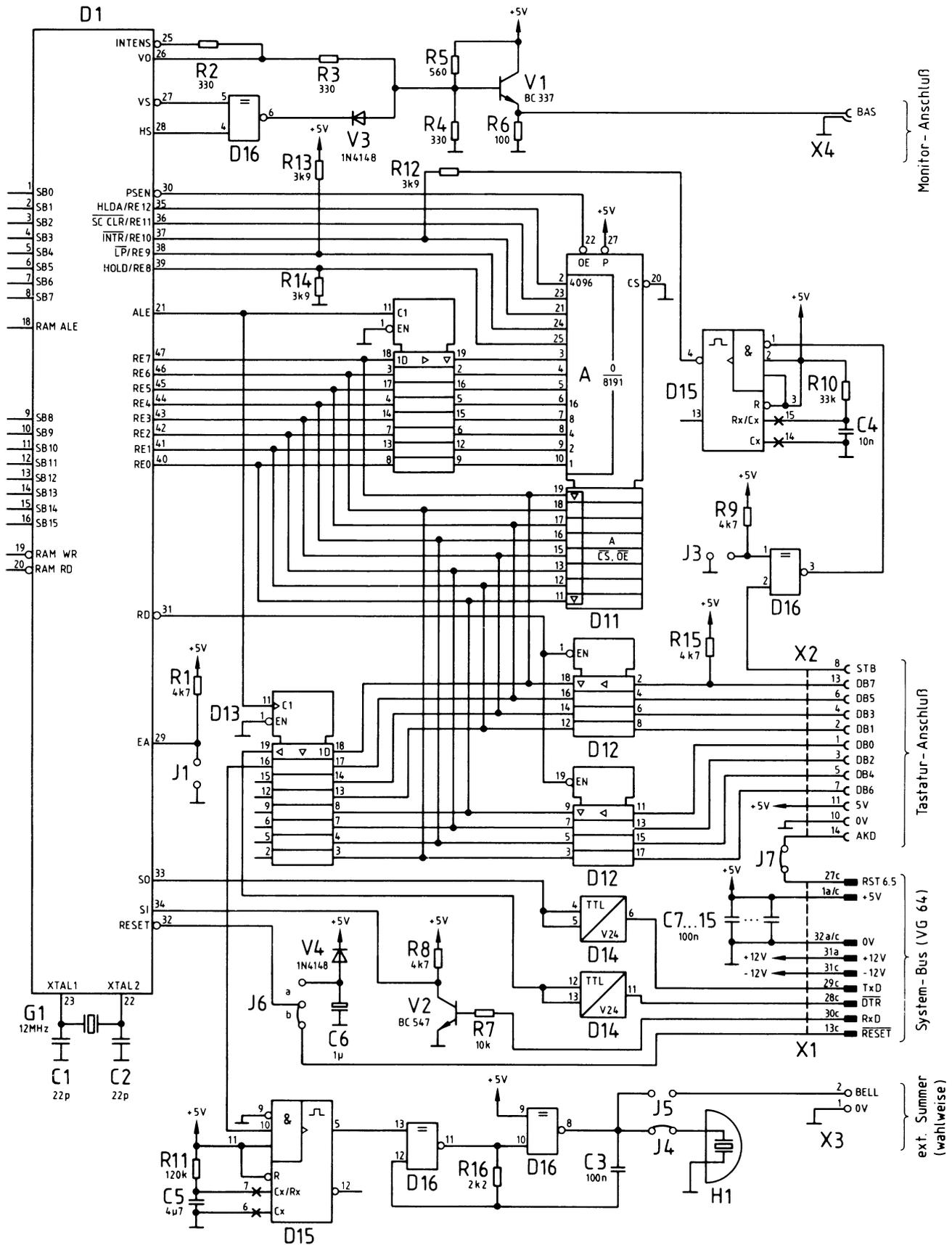
Anhang 1: Stromlaufplan



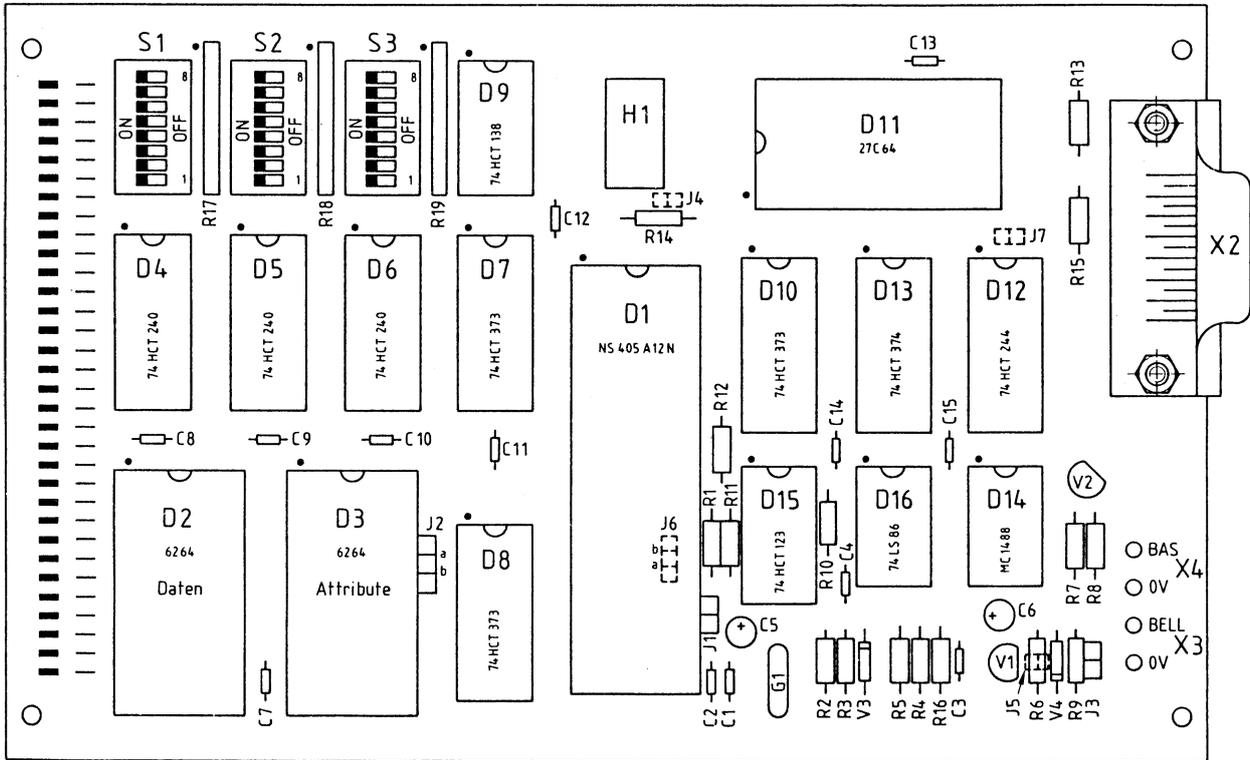
	D1	D2, D3	D4...D6	D7, D8, D10	D9	D11	D12	D13	D14	D15	D16
	NS405A12N	6264	74 HCT 240	74 HCT 373	74 HCT 138	27C64	74 HCT 244	74 HCT 374	MC 1488	74 HCT 123	74 LS 86
0V	24	14	10	10	8	14	10	10	7	8	7
+5V	48	28	20	20	16	1,28	20	20	-	16	14
+12V	-	-	-	-	-	-	-	-	14	-	-
-12V	-	-	-	-	-	-	-	-	1	-	-

R17... R19: SIL - Widerstandnetzwerk 4k7

Anhang 1: Stromlaufplan



Anhang 2: Bestückungsplan



Anhang 3: Bauteilliste

elektr. Bez.	Art	Bemerkung
H1	Piezokeramischer Signalgeber B/C 11-S	
C1, C2	Keramik-Kondensator 22 pF, 50 V	
C3, C7...C15	Keramik-Kondensator 0,1 uF, 50 V	
C4	Keramik-Kondensator 10 nF, 50 V	
C5	Tantal-Elko 4,7uF; 35V	Tropfenform
C6	Tantal-Elko 1uF; 35V	Tropfenform
D1	Terminal Management Prozessor NS 405 A12N	
D2, D3	RAM-Baustein 8K x 8 Bit, statisch max. 300ns, pinkompatibel mit 6264	
D4...D6	8 inv. Bus-Leitungstreiber 74 HCT 240	Tri-State
D7, D8, D10	8-Bit-D-Latch 74 HCT 373	Tri-State
D9	3-Bit-Binärdekoder/Demultiplexer 74 HCT 138	
D11	EPROM-Baustein 8K x Bit, CMOS max. 300ns, pinkompatibel mit 2764	programmiert mit der Software des Video-Interfaces
D12	8 Bus-Leitungstreiber 74 HCT 244	Tri-State
D13	8-Bit-D-Register 74 HCT 374	Tri-State
D14	4 Leitungstreiber für V24-Schnittstellen MC 1488 oder 75188	
D15	2 retriggerbare Monoflops 74 HCT 123	
D16	4 Exklusiv-Oder-Gatter je 2 Eing. 74 LS 86	kein HCT verwenden!
G1	Quarz 12 MHz, HC18U stehend	
R1, R8, R9, R15	Widerstand 4,7 kOhm	Alle Widerstände 0,25W; ±5%
R2, R3, R4	Widerstand 330 Ohm	
R5	Widerstand 560 Ohm	
R6	Widerstand 100 Ohm	
R7	Widerstand 10 kOhm	
R10	Widerstand 33 kOhm	
R11	Widerstand 120 kOhm	
R12...R14	Widerstand 3,9 kOhm	
R16	Widerstand 2,2 kOhm	
R17...R19	SIL-Widerstandsnetzwerk 8x4,7 kOhm	1 gemeinsamer Anschluß
S1...S3	Miniatur-Schiebeschalter 8-polig, DIL	
V1	Si-NPN-Transistor BC 337-25	
V2	Si-NPN-Transistor BC 547-25	
V3, V4	Si-Diode 1N4148	

Anhang 4: Lötbrückentabelle

Auf der Video-Interface-Baugruppe befinden sich einige Lötbrücken (Jumper). Diese ermöglichen eine Anpassung der Hardware an bestimmte Erfordernisse.

- J1 - Umschaltung internes/externes ROM für den TMP
geöffnet: externes ROM
geschlossen: internes ROM (nur bei NS455!)
Bei dem hier verwendeten Typ (NS 405) immer offen !
Lieferzustand: geöffnet (externes ROM)
- J2 - Speichertyp-Umschaltung
Stellung "a": 6264 (8K x 8-Bit)
Stellung "b": 6116 (2K x 8-Bit)
Es darf immer nur eine Brücke ("a" oder "b") geschlossen sein!
Lieferzustand: Stellung "a" (6264, 8K x 8-Bit)
- J3 - Wahl des aktiven Pegels des Tastatur-STROBE-Signals
geöffnet: H-Pegel (MFA-Standard-Tastatur)
geschlossen: L-Pegel
Lieferzustand: geöffnet (H-Pegel)
- J4 - Interner Signalgeber H1 ein/aus
geöffnet: interner Signalgeber aus
geschlossen: interner Signalgeber ein
Lieferzustand: geschlossen (interner Signalgeber ein)
- J5 - An X3 anschließbarer externer Signalgeber ein/aus
geöffnet: externer Signalgeber aus
geschlossen: externer Signalgeber ein
Lieferzustand: geschlossen (externer Signalgeber ein)

Der in der Tastatur eingebaute Signalgeber kann nicht als externer Signalgeber verwendet werden! Als externe Signalgeber können nur piezokeramische Signalgeber verwendet werden.

Anhang 4: Lötbrückentabelle

J6 - Umschaltung externer/interner Reset

Stellung "a": interner Reset

Stellung "b": externer Reset

Es darf immer nur eine Brücke ("a" oder "b") geschlossen sein!

Lieferzustand: Stellung "b" (externer Reset)

Bei einem **externen** Reset wird das Video-Interface beim Betätigen des an der CPU-Baugruppe befindlichen Reset-Tasters zurückgesetzt. Dies kann sinnvoll sein, wenn versehentlich falsche Steuersequenzen zum Interface übertragen wurden und der Anwender nicht weiß, mit welchen Steuersequenzen er den Ursprungszustand wieder herstellen kann. In manchen Fällen kann es allerdings nachteilig sein, daß beim Rücksetzen der CPU auch das Video-Interface zurückgesetzt wird, da es dann die über die DIL-Schalter festgelegte Betriebsart einnimmt. Alle Änderungen der Betriebsart, die eventuell zuvor über Steuersequenzen vorgenommen wurden, werden damit hinfällig.

J7 - AKD-Signal der Tastatur zum RST 6.5-Anschluß der CPU durchschleifen

geöffnet: AKD-Signal **nicht** durchschleifen

geschlossen: AKD-Signal durchschleifen

Lieferzustand: geschlossen (AKD-Signal auf RST 6.5-Anschluß)

Bei geschlossener Brücke erhält die CPU bei jedem Tastendruck eine Interrupt-Anforderung. Dadurch kann ein laufendes BFZ-STEUER-BASIC- oder SPS-Programm durch Betätigung einer Taste (außer SHIFT, CONTROL, BREAK) abgebrochen werden. Bei geöffneter Brücke können diese Programme nur durch die Betätigung des Reset-Tasters abgebrochen werden.

Anhang 5: Steckerbelegungen

X1 - MFA-Systembus (VG64-Steckerleiste)

Kontakt	Signalbezeichnung	
1 a/c	5V	Betriebsspannung
27 c	RST 6.5	AKD-Signal der Tastatur
28 c	DTR	Data Terminal Ready (Steuersignal)
29 c	TxD	Daten vom Video-Interface
30 c	RxD	Daten zum Video-Interface
31 a	+12V	Betriebsspannung
31 c	-12V	Betriebsspannung
32 a/c	0V	Masse (Ground)

X2 - Tastatur-Anschluß (15-polige Sub-D-Buchse)

Kontakt	Signalbezeichnung	
1	DB0	Tastatur-Datenleitung 0
2	DB1	Tastatur-Datenleitung 1
3	DB2	Tastatur-Datenleitung 2
4	DB3	Tastatur-Datenleitung 3
5	DB4	Tastatur-Datenleitung 4
6	DB5	Tastatur-Datenleitung 5
7	DB6	Tastatur-Datenleitung 6
8	STB	Tastatur-STROBE
9	---	Nicht belegt
10	0V	Masse (Ground)
11	5V	Betriebsspannung der Tastatur
12	---	Nicht belegt
13	DB7	Tastatur-Datenleitung 7 (falls angeschl.)
14	AKD	Any Key Down
15	---	Nicht belegt

X3 - Anschluß für ext. Signalgeber (2 Lötunkte)

Kontakt	Signalbezeichnung	
1	0V	Masse (Ground)
2	BELL	Signalgeber-Ansteuerung

X4 - Monitor-Anschluß (BNC-Buchse)

Kontakt	Signalbezeichnung	
Mitte	BAS	Bild-, Austast- und Synchron-Signal
Schirm	0V	Masse (Ground)

Anhang 6: ASCII-Tabelle

Höherw. Niederw.		0	1	2	3	4	5	6	7
		0000	0001	0010	0011	0100	0101	0110	0111
0	0000	NUL 0	DLE 16	Space 32	0 48	@ 64	P 80	' 96	p 112
1	0001	SOH 1	DC1 17	! 33	1 49	A 65	Q 81	a 97	q 113
2	0010	STX 2	DC2 18	" 34	2 50	B 66	R 82	b 98	r 114
3	0011	ETX 3	DC3 19	# 35	3 51	C 67	S 83	c 99	s 115
4	0100	EOT 4	DC4 20	\$ 36	4 52	D 68	T 84	d 100	t 116
5	0101	ENQ 5	NAK 21	% 37	5 53	E 69	U 85	e 101	u 117
6	0110	ACK 6	SYN 22	& 38	6 54	F 70	V 86	f 102	v 118
7	0111	BEL 7	ETB 23	' 39	7 55	G 71	W 87	g 103	w 119
8	1000	BS 8	CAN 24	(40	8 56	H 72	X 88	h 104	x 120
9	1001	HT 9	EM 25) 41	9 57	I 73	Y 89	i 105	y 121
A	1010	LF 10	SUB 26	* 42	: 58	J 74	Z 90	j 106	z 122
B	1011	VT 11	ESC 27	+ 43	; 59	K 75	[91	k 107	{ 123
C	1100	FF 12	FS 28	, 44	< 60	L 76	\ 92	l 108	! 124
D	1101	CR 13	GS 29	- 45	= 61	M 77] 93	m 109	} 125
E	1110	SO 14	RS 30	. 46	> 62	N 78	^ 94	n 110	~ 126
F	1111	SI 15	US 31	/ 47	? 63	O 79	_ 95	o 111	⋯ 127

Anhang 7: Tabelle der länderspezifischen Zeichen

ASCII-Code		Nationalitäten							
hex.	dez.	USA	FRA	GER	ENG	DEN	SWE	ITA	SPA
23	35	#	#	#	£	#	#	#	#
24	36	\$	\$	\$	\$	\$	α	\$	\$
40	64	@	à	§	@	@	@	@	@
5B	91	[°	Ä	[Æ	Ä	°	ì
5C	92	\	ç	Ö	\	ø	Ö	\	Ñ
5D	93]	§	Ü]	Å	Å	é]
5E	94	^	^	^	^	^	Ü	^	^
60	96	`	`	`	`	`	é	ù	`
7B	123	{	é	ä	{	æ	ä	à	¨
7C	124		ù	ö		ø	ö	ö	ñ
7D	125	}	è	ü	}	å	å	è	}
7E	126	~	¨	ß	~	~	ü	ì	~

Anhang 8: Darstellung der Steuerzeichen im Control-Mode

Im Control-Mode werden alle **Steuerzeichen**, die das Video-Interface empfängt, nicht ausgeführt, sondern als inverse Zeichen mit halber Helligkeit auf dem Bildschirm angezeigt. In der nachstehenden Tabelle sind Code und Mnemonik der Steuerzeichen sowie angezeigte Zeichen gegenübergestellt.

Zwischen den Steuerzeichen und den angezeigten Zeichen besteht folgender Zusammenhang:

Code des angezeigten Zeichens = Code des Steuerzeichens + 40_H

Die Codes der Steuerzeichen lassen sich auch von der Tastatur zum Mikrocomputer übertragen. Hierzu muß das in der rechten Spalte der Tabelle aufgeführte Zeichen ("angezeigtes Zeichen") bei gleichzeitiger Betätigung der CONTROL-Taste eingegeben werden.

Beispiel:

Soll z.B. das Steuerzeichen NUL (Code 00_H) zum Mikrocomputer übertragen werden, ist die CONTROL-Taste zusammen mit der Taste "@" zu betätigen.

Steuerzeichen		angezeigtes Zeichen ¹⁾	Steuerzeichen		angezeigtes Zeichen ¹⁾
Code	Mnemonik		Code	Mnemonik	
00 _H	NUL	@	10 _H	DLE	P
01 _H	SOH	A	11 _H	DC1	Q
02 _H	STX	B	12 _H	DC2	R
03 _H	ETX	C	13 _H	DC3	S
04 _H	EOT	D	14 _H	DC4	T
05 _H	ENQ	E	15 _H	NAK	U
06 _H	ACK	F	16 _H	SYN	V
07 _H	BEL	G	17 _H	ETB	W
08 _H	BS	H	18 _H	CAN	X
09 _H	HT	I	19 _H	EM	Y
0A _H	LF	J	1A _H	SUB	Z
0B _H	VT	K	1B _H	ESC	[
0C _H	FF	L	1C _H	FS	\
0D _H	CR	M	1D _H	GS]
0E _H	SO	N	1E _H	RS	^
0F _H	SI	O	1F _H	US	_

¹⁾ Die angezeigten Zeichen sind abhängig vom eingestellten länderspezifischen Zeichensatz. Für diese Tabelle wurde der USA-Zeichensatz (empfohlene Grundeinstellung und Tastaturbeschriftung) verwendet.

Anhang 9: Bedeutung der Schalterstellungen der Schalter-Reihen S1 bis S3

Auf der Baugruppe befinden sich drei Schalter-Reihen. Jede dieser Schalter-Reihen besteht aus acht Schaltern. Die einzelnen Schalter-Reihen sind mit S1, S2 und S3 bezeichnet. S1.1 bezeichnet den Schalter Nummer 1 in der Schalter-Reihe S1. Die anderen Schalter sind entsprechend nummeriert. Mit Hilfe dieser Schalter kann die Grundeinstellung des Video-Interfaces vorgenommen werden. Diese Grundeinstellung wird nach dem Einschalten des Interfaces aktiv. Die Grundeinstellung kann aber während des Betriebes durch die Übertragung entsprechender Steuerzeichen bzw. durch das Set-Up-Menü verändert werden. Die folgende Übersicht zeigt die Einstellmöglichkeiten dieser Schalter-Reihen:

- Schalter-Reihe S1:

S1.1 (Emulations-Mode):

S1.1	Emulations-Mode
OFF	TVI 950
ON	MAT 85

S1.2 (Cursor-Art):

S1.2	Cursor-Art
OFF	blinkend
ON	ruhend

S1.3 (Cursor-Form):

S1.3	Cursor-Form
OFF	Strich
ON	Block

S1.4 (Bildhintergrund):

S1.4	Bildhintergrund
OFF	dunkel
ON	hell

S1.5 (Reserviert, muß immer in Stellung OFF sein)

Anhang 9: Bedeutung der Schalterstellungen der Schalter-Reihen S1 bis S3

S1.6 (Handshake):

S1.6	Handshake
OFF	Hardware (DTR)
ON	Software (XON/XOFF)

S1.7 (Reserviert, muß immer in Stellung OFF sein)

S1.8 (Statuszeile):

S1.8	Statuszeile
OFF	aus
ON	ein

- Schalter-Reihe S2

S2.1 (Wortlänge):

S2.1	Wortlänge
OFF	7 Bit
ON	8 Bit

S2.2 (Zeichensatz):

S2.3

S2.4

S2.2	S2.3	S2.4	Zeichensatz
OFF	OFF	OFF	USA
ON	OFF	OFF	Frankreich
OFF	ON	OFF	Deutschland
ON	ON	OFF	England
OFF	OFF	ON	Dänemark
ON	OFF	ON	Schweden
OFF	ON	ON	Italien
ON	ON	ON	Spanien

Anhang 9: Bedeutung der Schalterstellungen der Schalter-Reihen S1 bis S3

S2.5 (Zeilenanzahl):

S2.6

S2.5	S2.6	Zeilenanzahl
OFF	OFF	22
ON	OFF	24
OFF	ON	26
ON	ON	28

S2.7 (Spaltenanzahl):

S2.8

S2.7	S2.8	Spaltenanzahl
OFF	OFF	72
ON	OFF	80
OFF	ON	88
ON	ON	96

- Schalter-Reihe S3

S3.1 (Anzahl der Stopbits):

S3.1	Stopbits
OFF	1 Bit
ON	2 Bit

S3.2 (Paritätsbit):

S3.3

S3.4

S3.2	S3.3	S3.4	Paritätsbit
OFF	OFF	OFF	kein Paritätsbit, no Parity
ON	OFF	OFF	ungerade Parität, odd Parity
OFF	ON	OFF	kein Paritätsbit, no Parity
ON	ON	OFF	gerade Parität, even Parity
OFF	OFF	ON	kein Paritätsbit, no Parity
ON	OFF	ON	immer "1", Mark Parity
OFF	ON	ON	kein Paritätsbit, no Parity
ON	ON	ON	immer "0", Space Parity

Anhang 9: Bedeutung der Schalterstellungen der Schalter-Reihen S1 bis S3

S3.5 (Baud-Rate):

S3.6

S3.7

S3.8

S3.5	S3.6	S3.7	S3.8	Baud-Rate
OFF	OFF	OFF	OFF	110 Bd
ON	OFF	OFF	OFF	134,5 Bd
OFF	ON	OFF	OFF	150 Bd
ON	ON	OFF	OFF	300 Bd
OFF	OFF	ON	OFF	600 Bd
ON	OFF	ON	OFF	1200 Bd
OFF	ON	ON	OFF	1800 Bd
ON	ON	ON	OFF	2400 Bd
OFF	OFF	OFF	ON	3600 Bd
ON	OFF	OFF	ON	4800 Bd
OFF	ON	OFF	ON	7200 Bd
ON	ON	OFF	ON	9600 Bd
OFF	OFF	ON	ON	19200 Bd
ON	OFF	ON	ON	19200 Bd
OFF	ON	ON	ON	19200 Bd
ON	ON	ON	ON	9600 Bd

Anhang 10: Tabelle der Attribut-Bytes für die Escape-Sequenz "ESC G"

Mit Hilfe der Escape-Sequenz "ESC G" können den einzelnen Zeichen im TVI 950-Mode des Interfaces verschiedene Attribute zugewiesen werden. Hierzu ist nach den beiden Bytes für die einleitenden Zeichen "ESC G" ein drittes Byte zu übertragen. Mit diesem dritten Byte "Attribute+30_H" werden die gewünschten Attribute festgelegt. Die Tabelle auf der nächsten Seite soll helfen, dieses Byte zu ermitteln.

Im rechten Tabellenteil findet man die einzelnen Attribute: "unsichtbar", "blinkend", "invers", "unterstrichen", "doppelt breit" und "doppelt hoch". Aktive Attribute sind in der entsprechenden Spalte durch das Symbol "*" gekennzeichnet. Zur Ermittlung des dritten Bytes der Escape-Sequenz sucht man die Tabellen-Zeile auf, in der die gewünschte Attribut-Kombination durch "*" -Symbole gekennzeichnet ist. Im linken Tabellenteil findet man das entsprechende Byte (jeweils in ASCII-, Hexadezimal- und Dezimalcodierung). Die Tabelle enthält alle zulässigen Attribut-Kombinationen.

Beispiel:

Es soll die Attribut-Kombination "unterstrichen" / "blinkend" eingeschaltet werden. In der Tabelle ist hierfür folgende Zeile gültig:

3. Byte "Attribute+30 _H "			Bits						
			5 doppelt hoch	4 doppelt breit	3 unterstr.	2 invers	1 blinkend	0 unsichtbar	
ASCII ¹⁾	hex.	dez.							
:	3A	58	-	-	*	-	*	-	

Aus der Tabelle entnimmt man als drittes Byte der Escape-Sequenz das ASCII-Zeichen ":". Hiermit ergibt sich folgende Gesamtsequenz:

```

ESC G   :   (ASCII codiert)
1BH 47H 3AH (hex. codiert)
27D 71D 58D (dez. codiert)

```

Anhang 10: Tabelle der Attribut-Bytes für die Escape-Sequenz "ESC G"

3. Byte "Attribute+30 _H "			Bits					
			5 doppelt hoch	4 doppelt breit	3 unterstr.	2 invers	1 blinkend	0 unsichtbar
ASCII ¹⁾	hex.	dez.						
0	30	48	-	-	-	-	-	-
1	31	49	-	-	-	-	-	*
2	32	50	-	-	-	-	*	-
3	33	51	-	-	-	-	*	*
4	34	52	-	-	-	*	-	-
5	35	53	-	-	-	*	-	*
6	36	54	-	-	-	*	*	-
7	37	55	-	-	-	*	*	*
8	38	56	-	-	*	-	-	-
9	39	57	-	-	*	-	-	*
:	3A	58	-	-	*	-	*	-
;	3B	59	-	-	*	-	*	*
<	3C	60	-	-	*	*	-	-
=	3D	61	-	-	*	*	-	*
>	3E	62	-	-	*	*	*	-
?	3F	63	-	-	*	*	*	*
@	40	64	-	*	-	-	-	-
A	41	65	-	*	-	-	-	*
B	42	66	-	*	-	-	*	-
C	43	67	-	*	-	-	*	*
D	44	68	-	*	-	*	-	-
E	45	69	-	*	-	*	-	*
F	46	70	-	*	-	*	*	-
G	47	71	-	*	-	*	*	*
H	48	72	-	*	*	-	-	-
I	49	73	-	*	*	-	-	*
J	4A	74	-	*	*	-	*	-
K	4B	75	-	*	*	-	*	*
L	4C	76	-	*	*	*	-	-
M	4D	77	-	*	*	*	*	*
N	4E	78	-	*	*	*	*	-
O	4F	79	-	*	*	*	*	*
P	50	80	*	-	-	-	-	-
R	52	82	*	-	-	-	*	-
T	54	84	*	-	-	*	-	-
V	56	86	*	-	-	*	*	-
X	58	88	*	-	*	-	-	-
Z	5A	90	*	-	*	-	*	-
\	5C	92	*	-	*	*	-	-
^	5E	94	*	-	*	*	*	-
'	60	96	*	*	-	-	-	-
b	62	98	*	*	-	-	*	-
d	64	100	*	*	-	*	-	-
f	66	102	*	*	-	*	*	-
h	68	104	*	*	*	-	-	-
j	6A	106	*	*	*	-	*	-
l	6C	108	*	*	*	*	-	-
n	6E	110	*	*	*	*	*	-

¹⁾ Für die Darstellung der ASCII-Zeichen wurde der USA-Zeichensatz (empfohlene Grundeinstellung und Tastaturbeschriftung) verwendet.

Anhang 11: Steuerzeichen im MAT 85-Mode

- 1-Byte-Steuerzeichen

Funktion	Bezeichnung	hex. Code	dez. Code	Seite
Bell	BEL	07	7	15
Cursor nach links (Backspace)	BS	08	8	15
Cursor nach rechts	HT	09	9	15
Zeilenvorschub (Cursor nach unten)	LF	0A	10	16
Cursor nach oben	VT	0B	11	16
Bildschirm löschen, Cursor nach oben links	FF	0C	12	16
Wagenrücklauf, Zeilenrest löschen	CR	0D	13	16
Zeile löschen	SUB	1A	26	16
Cursor nach unten	ESC	1B	27	17
Cursor nach oben links	FS	1C	28	17
Cursor an den Zeilenanfang	GS	1D	29	17

- DLE-Sequenzen

Funktion	Bezeichnung	hex. Code	dez. Code	Seite
Modeumschaltung (TVI 950)	DLE DLE 1	10 10 31	16 16 49	17
Modeumschaltung (MAT 85)	DLE DLE 2	10 10 32	16 16 50	18
Modeabfrage	DLE DLE ?	10 10 3F	16 16 63	18
Software-Reset	DLE DLE @	10 10 40	16 16 64	18
Versionsabfrage	DLE DLE V	10 10 56	16 16 86	18

Anhang 12: Steuerzeichen im TVI 950-Mode

- 1-Byte-Steuerzeichen

Funktion	Bezeichnung	hex. Code	dez. Code	Seite
Bell	BEL	07	7	19
Cursor nach links (Backspace)	BS	08	8	19
Horizontal-Tabulator	HT	09	9	20
Zeilenvorschub (Cursor nach unten)	LF	0A	10	20
Cursor nach oben	VT	0B	11	20
Cursor nach rechts	FF	0C	12	20
Wagenrücklauf (Cursor an den Zeilenanfang)	CR	0D	13	21
Cursor nach unten	SYN	16	22	21
Bildschirm löschen, Cursor nach oben links	SUB	1A	26	21
Cursor nach oben links	RS	1E	30	21

- DLE-Sequenzen

Funktion	Bezeichnung	hex. Code	dez. Code	Seite
Modeumschaltung (TVI 950)	DLE DLE 1	10 10 31	16 16 49	31
Modeumschaltung (MAT 85)	DLE DLE 2	10 10 32	16 16 50	31
Modeabfrage	DLE DLE ?	10 10 3F	16 16 63	31
Software-Reset	DLE DLE @	10 10 40	16 16 64	31
Versionsabfrage	DLE DLE V	10 10 56	16 16 86	32

Anhang 12: Steuerzeichen im TVI 950-Mode

- Escape-Sequenzen (nach Funktionen geordnet)

Erfordert eine Escape-Sequenz zusätzliche Parameter, so ist hier der Platzhalter "_" angegeben.

Funktion	Bezeichnung	hex. Code	dez. Code	Seite
<u>Bildschirm löschen, Cursor nach oben links</u>				
Bildschirm löschen, Cursor nach oben links	ESC *	1B 2A	27 42	22
Bildschirm löschen, Cursor nach oben links	ESC +	1B 2B	27 43	22
Bildschirm löschen, Cursor nach oben links	ESC ,	1B 2C	27 44	22
Bildschirm löschen, Cursor nach oben links	ESC :	1B 3A	27 58	22
Bildschirm löschen, Cursor nach oben links	ESC ;	1B 3B	27 59	22
<u>Schriftfeld editieren</u>				
Leerzeichen einfügen	ESC Q	1B 51	27 81	22
Zeichen löschen	ESC W	1B 57	27 87	22
Zeile einfügen	ESC E	1B 45	27 69	22
Zeile löschen	ESC R	1B 52	27 82	23
Zeile ab Cursor-Position löschen	ESC T	1B 54	27 84	23
Zeile ab Cursor-Position löschen	ESC t	1B 74	27 116	23
Schriftfeld ab Cursor-Position löschen	ESC Y	1B 59	27 89	23
Schriftfeld ab Cursor-Position löschen	ESC y	1B 79	27 121	23
<u>Cursor-Positionierung</u>				
Cursor-Position setzen	ESC = _ _	1B 3D _ _	27 61 _ _	23
Cursor-Position abfragen	ESC ?	1B 3F	27 63	24
Cursor nach links zum nächsten Tabulator (Backtab)	ESC I	1B 49	27 73	25
<u>Zeichensätze, Zeichen- und Cursor-Darstellung</u>				
Zeichensatz wählen	ESC z _	1B 7A _	27 122 _	25
Attribute setzen/löschen	ESC G _	1B 47 _	27 71 _	27
Halbe Helligkeit	ESC)	1B 29	27 41	29
Volle Helligkeit	ESC (1B 28	27 40	29
Bildhintergrund hell	ESC b	1B 62	27 98	29
Bildhintergrund dunkel	ESC d	1B 64	27 100	29
Control-Mode ein	ESC U	1B 55	27 85	29
Cursor-Darstellung	ESC . _	1B 2E _	27 46 _	30

Anhang 12: Steuerzeichen im TVI 950-Mode

- Escape-Sequenzen (nach Codes geordnet)

Erfordert eine Escape-Sequenz zusätzliche Parameter, so ist hier der Platzhalter "_" angegeben.

Funktion	Bezeichnung	hex. Code	dez. Code	Seite
Volle Helligkeit	ESC (1B 28	27 40	29
Halbe Helligkeit	ESC)	1B 29	27 41	29
Bildschirm löschen, Cursor nach oben links	ESC *	1B 2A	27 42	22
Bildschirm löschen, Cursor nach oben links	ESC +	1B 2B	27 43	22
Bildschirm löschen, Cursor nach oben links	ESC ,	1B 2C	27 44	22
Cursor-Darstellung	ESC . _	1B 2E _	27 46 _	30
Bildschirm löschen, Cursor nach oben links	ESC :	1B 3A	27 58	22
Bildschirm löschen, Cursor nach oben links	ESC ;	1B 3B	27 59	22
Cursor-Position setzen	ESC = _ _	1B 3D _ _	27 61 _ _	23
Cursor-Position abfragen	ESC ?	1B 3F	27 63	24
Zeile einfügen	ESC E	1B 45	27 69	22
Attribute setzen/löschen	ESC G _	1B 47 _	27 71 _	27
Cursor nach links zum nächsten Tabulator (Backtab)	ESC I	1B 49	27 73	25
Leerzeichen einfügen	ESC Q	1B 51	27 81	22
Zeile löschen	ESC R	1B 52	27 82	23
Zeile ab Cursor-Position löschen	ESC T	1B 54	27 84	23
Control-Mode ein	ESC U	1B 55	27 85	29
Zeichen löschen	ESC W	1B 57	27 87	22
Schriftfeld ab Cursor-Position löschen	ESC Y	1B 59	27 89	23
Bildhintergrund hell	ESC b	1B 62	27 98	29
Bildhintergrund dunkel	ESC d	1B 64	27 100	29
Zeile ab Cursor-Position löschen	ESC t	1B 74	27 116	23
Schriftfeld ab Cursor-Position löschen	ESC y	1B 79	27 121	23
Zeichensatz wählen	ESC z _	1B 7A _	27 122 _	25

Anhang 13: Stichwortverzeichnis

A

Anpassung des CP/M-Editors BFZED.COM	58ff
ASCII	8f, 25f, 75f
Attribute	8, 12f, 27f, 64ff, 82f
Attributlogik	13
Attribut-Speicher	8, 12f
Austast-Impulse	12

B

Backspace	15, 19, 84f.
BASIC	39ff, 55ff
BAS-Signal	6f, 12
Baudrate	35, 43ff, 81
BDOS-Funktion	48ff
Betriebsprogramm MAT 85	33ff
Betriebssystem CP/M	43ff
BFZED.COM, Anpassung des CP/M-Editors ...	58ff
BFZ-STEUER-BASIC	39ff
Bilddarstellung	5f, 10
Bildwiederhol-Speicher	8f, 13

C

Carriage Return	12, 16f, 21, 84f
CCIR-Fernsehnorm	6
Character-Generator	8, 11, 13
Control-Mode	29f, 67, 77, 86f
CP/M	4, 43ff
CRT-Controller	8ff, 13
Cursor	10, 14
Cursor-Darstellung	14, 30, 86ff
Cursor-Position	15ff, 19ff, 23ff, 84ff

D

Darstellung von Zeichen	5f, 10ff, 25ff, 75ff
Datensichtsanion	4f
DIL-Schalter	4, 34f, 46f, 78ff
DLE-Sequenzen	17f, 31f, 84f
DSR	43ff
DTR	45f

E

Editor BFZED.COM (Anpassung)	58ff
Emulation	14, 19, 62, 64f, 78
Escape-Sequenzen	19, 21ff, 82f, 86f

Anhang 13: Stichwortverzeichnis

H

Handshake	43, 45
Horizontal-Synchronimpulse	5ff, 12

K

Kompatibilität zum Video-Interface BFZ/MFA 8.2	4, 14ff
--	---------

L

Länderspezifischer Zeichensatz	25f, 34, 47, 76, 79, 86f
Line Feed	10, 16, 20, 84f
Lötbrücken	43ff, 72f

M

MAT 85-Mode	14ff, 84
Matrixraster	5, 11
MBASIC	55ff
Mikrocontroller	8ff
Mischstufe	12f

P

Pixel	5f
Programmierbare Serienschnittstelle	43ff
Punktzeilen-Daten	5ff, 11f

R

Reihen-Adresse	11
----------------	----

S

Sägezahngenerator	5
Schieberegister	12f
Schriftfeld	6, 8f
Set-Up-Menü	64ff
Signalgeber	4, 72, 74
Statuszeile	34, 46, 62f, 79
STEUER-BASIC	39ff
Steuerzeichen	10, 14ff, 75, 84ff
String	48ff, 56
Synchron-Impulse	5ff, 12

T

Tabulator	20, 25, 85ff
Tastatur	4, 26
TeleVideo	14
Terminal	14
TVI 950-Mode	4, 14, 19ff

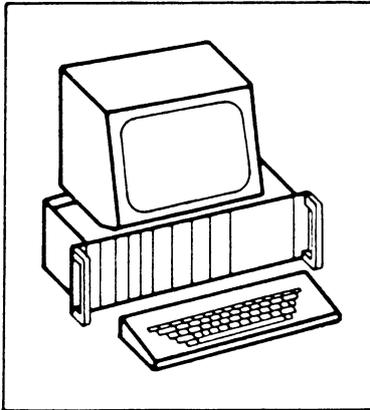
Anhang 13: Stichwortverzeichnis

V	
V24/RS232-Schnittstelle	43f
Vertikal-Synchronimpulse	6, 12
Video-Interface BFZ/MFA 8.2, Kompatibilität zum ...	4, 14ff
W	
Wagenrücklauf	12, 16f, 21, 84f
Z	
Zeichen, Darstellung von ...	5f, 10ff, 25ff, 75ff
Zeichen-Adresse	11
Zeichencode-Speicher	8f, 12f
Zeichenform	8, 11
Zeichengenerator	8, 11
Zeichensatz, länderspezifischer ...	25f, 34, 47, 76f, 79, 86f
Zeichenspalten-Zähler	10
Zeichenzeilen-Zähler	10
Zeilenvorschub	10, 16, 20, 84f

Anhang 14: Verzeichnis der Abbildungen

Bild 1:	Datensichtstation und Mikrocomputer	4
Bild 2:	Zeichendarstellung im 5x9-Raster	5
Bild 3:	Schriftfeld auf dem Bildschirm	6
Bild 4:	Das BAS-Signal	7
Bild 5:	Blockschaltbild des Video-Interfaces	9
Bild 6:	Programmierung des Zeichengenerators (Auszug)	11
Bild 7:	Datenaustausch zwischen CPU und Video-Interface (MAT 85)	33
Bild 8:	Bestückungsplan der V24/RS232-Schnittstelle	44
Bild 9:	Bestückungsplan der Programmierbaren Serienschnittstelle	45
Bild 10:	Datenaustausch zwischen CPU und Video-Interface (CP/M)	45
	Stromlaufplan	68
	Bestückungsplan	70

FACHPRAKTISCHE ÜBUNG MIKROCOMPUTER-TECHNIK



Schnelleres
Video-Interface



1. Leistungsbeschreibung der Umrüstung "MAT85 V1.8/S"

Die Umrüstung "MAT85 V1.8/S" erhöht die Geschwindigkeit der Zeichenübertragung zwischen CPU-Baugruppe und dem Video-Interface BFZ/MFA-8.2 bzw. BFZ/MFA-8.4 von 1200 Bit/s auf 9600 Bit/s, also um den Faktor 8.

Dieses wird zum einen durch Änderung der "Seriellen Ein-/Ausgabe-Routinen" des Betriebsprogramms MAT85 erreicht. Diese Software-Anpassung ist im mitgelieferten EPROM "MAT85 V1.8/S" enthalten.

Zusätzlich sind auf dem Video-Interface geringfügige Änderungen vorzunehmen:

Beim Video-Interface BFZ/MFA-8.2 muß der Baudraten-Generator in seiner Taktfrequenz verändert werden (UART-Schiebetakt = 154 kHz). Dieses geschieht durch das Auftrennen zweier Leiterbahnen sowie dem Einlöten von zwei zusätzlichen Drahtbrücken auf dieser Baugruppe (siehe "Installation und Inbetriebnahme").

Die Änderungen beim Video-Interface BFZ/MFA-8.4 beschränken sich auf eine veränderte Einstellung der Schalterreihen S2 und S3 (siehe "Installation und Inbetriebnahme").

Durch die Umrüstung des Betriebsprogramms auf "MAT85 V1.8/S" ergeben sich erheblich kürzere Zeiten für die Übertragung von Ausgabezeichen auf den Bildschirm. Dieses bedeutet für Anwendungen, bei denen viele Zeichen zum Bildschirm ausgegeben werden, einen erheblichen Bedienkomfort (z.B. EDITOR).

In welchen Größenordnungen die Zeitersparnis zur Bildschirmausgabe liegen, soll durch ein Berechnungsbeispiel dargestellt werden:

Zur seriellen Übertragung eines Datenbytes müssen neben den 8 Datenbit zusätzlich 1 Startbit und 2 Stopbit übertragen werden, also 11 Bit/Zeichen. Damit ergeben sich für die zu vergleichenden Übertragungsraten folgende Übertragungszeiten für ein komplettes Zeichen:

1200 Baud:

9600 Baud:

$11/1200 \text{ s} = 9,17 \text{ ms}$

$11/9600 \text{ s} = 1,15 \text{ ms}$

Eine "volle" Bildschirmseite besteht z.B. aus 16 Zeilen mit je 64 Zeichen, das sind insgesamt 1024 Zeichen.

Zusätzlich müssen nach jeder Zeile die Steuerzeichen "CARRIAGE RETURN" (CR) und "LINE FEED" (LF) zum Video-Interface übertragen werden: 15 x CR, 15 x LF.

Es ergibt sich eine Gesamtzahl von 1054 Zeichen, so daß zur Übertragung folgende Zeiten benötigt werden:

MAT85 V1.8:
(1200 Baud)

MAT85 V1.8/S:
(9600 Baud)

$1054 \times 9,17 \text{ ms} = 9,67 \text{ s}$
=====

$1054 \times 1,15 \text{ ms} = 1,21 \text{ s}$
=====

Da das Video-Interface zur Verarbeitung einiger Steuerzeichen zusätzliche Zeit benötigt, durchläuft die Programmänderung "MAT85 V1.8/S" nach der Übertragung dieser Zeichen eine zusätzliche Zeitverzögerung:

Steuerzeichen		Hex	zus. Zeitverz.
"LINE FEED"	(Zeilenvorschub)	0A	9,2 ms
"CARRIAGE RETURN"	(mit Löschen Zeilenrest)	0D	2,7 ms
"CARRIAGE RETURN"	(ohne Löschen Zeilenrest)	1D	" "
"LINE DELETE"	(Zeile löschen)	1A	" "

Damit wird zur Übertragung dieser Steuerzeichen eine längere Zeit benötigt, als für andere Zeichen. Da allerdings in den meisten Anwendungen der Anteil der genannten Steuerzeichen verhältnismäßig gering ist zur Gesamtzeichenanzahl, wird die effektive Übertragungsrates nur geringfügig verringert (die "echte" Übertragungszeit für obiges Beispiel liegt dann bei 1,39 s).

2. Installation und Inbetriebnahme

Die Umrüstung "MAT85 V1.8/S" besteht aus einer Software-Änderung des Betriebsprogramms MAT85 und einer geringfügigen Hardware-Anpassung des Video-Interface BFZ/MFA-8.2 bzw. BFZ/MFA-8.4:

a) Software-Änderung

a1) Betriebsprogramm mit 2-K-EPROMs

Ersetzen Sie im Steckplatz 2 der EPROM-Karte für das Betriebsprogramm das ursprüngliche 2-K-EPROM

"MAT85 V1.8 II"

durch das mitgelieferte 2-K-EPROM

"MAT85 V1.8/S II"

a2) Betriebsprogramm mit 32-K-EPROM

Ersetzen Sie das 32-K-EPROM durch das mitgelieferte 32-K-EPROM mit der Änderung "MAT 32K/S".

b) Hardware-Änderung**b1) Video-Interface BFZ/MFA-8.2**

Führen Sie auf dem Video-Interface folgende Hardware-Anpassung durch:
(siehe auch Anlage: Hardware-Änderungen)

b1) Leiterbahnseite : Abtrennen Pin 14, IC 13
 " : Verbinden Pin 14, IC 13 mit Pin 14, IC 12
 " : Verbinden Pin 1, IC 12 mit Pin 40, IC 11

b2) Bestückungsseite: öffnen der Brücken P5 - P3 und P5 - P4

b2) Video-Interface BFZ/MFA-8.4

Die Schalterreihe S1 behält ihre ursprüngliche Schalterstellung:

Schalter	Stellung	eingestellte Funktion
S1.1	ON	MAT 85 - Mode
S1.2	OFF	blinkender Cursor
S1.3	OFF	Strich-Cursor
S1.4	OFF	dunkler Bildhintergrund
S1.5	OFF	reserviert, immer OFF!
S1.6	OFF	DTR-Handshake
S1.7	OFF	reserviert, immer OFF!
S1.8	OFF	Statuszeile aus

In der Schalterreihe S2 muß der Schalter S2.1 auf eine Wortlänge von 8 Bit eingestellt werden:

Schalter	Stellung	eingestellte Funktion
S2.1	ON	Wortlänge 8 Bit
S2.2	OFF	Zeichensatz USA
S2.3	OFF	dto.
S2.4	OFF	dto.
S2.5	ON	Zeilenzahl 24
S2.6	OFF	dto.
S2.7	ON	Spaltenzahl 80
S2.8	OFF	dto.

Mit der Schalterreihe S3 wird folgendes Übertragungsformat eingestellt:
9600 Bd, No Parity, 2 Stopbits:

Schalter	Stellung	eingestellte Funktion
S3.1	ON	2 Stopbit
S3.2	OFF	No Parity
S3.3	OFF	dto.
S3.4	OFF	dto.
S3.5	ON	9600 Bd
S3.6	ON	dto.
S3.7	OFF	dto.
S3.8	ON	dto.

Nach der Umrüstung läßt sich das MFA-System wie gewohnt in Betrieb nehmen. Es entfällt lediglich die ursprünglich erforderliche Betätigung der SPACE-Taste, da die Übertragungsrate zwischen CPU-Baugruppe und Video-Interface fest auf 9600 Baud initialisiert ist.

3. Hinweise zur Programmänderung "MAT85 V1.8/S"

Obwohl Änderungen in den "Seriellen Ein-/Ausgabe-Routinen" des Betriebsprogramms MAT85 vorgenommen wurden, sind die Einsprungsadressen für universell verwendbare Unterprogramme beibehalten worden, so daß diese Unterprogramme weiterhin uneingeschränkt benutzbar bleiben, z.B.:

- Tastaturabfrage	CALL 0043	oder	CALL 082D
- Bildschirmausgabe	CALL 0052	oder	CALL 089F
- Zeitschleife 0,24 s	CALL 0895		

4. Spezielle Hinweise

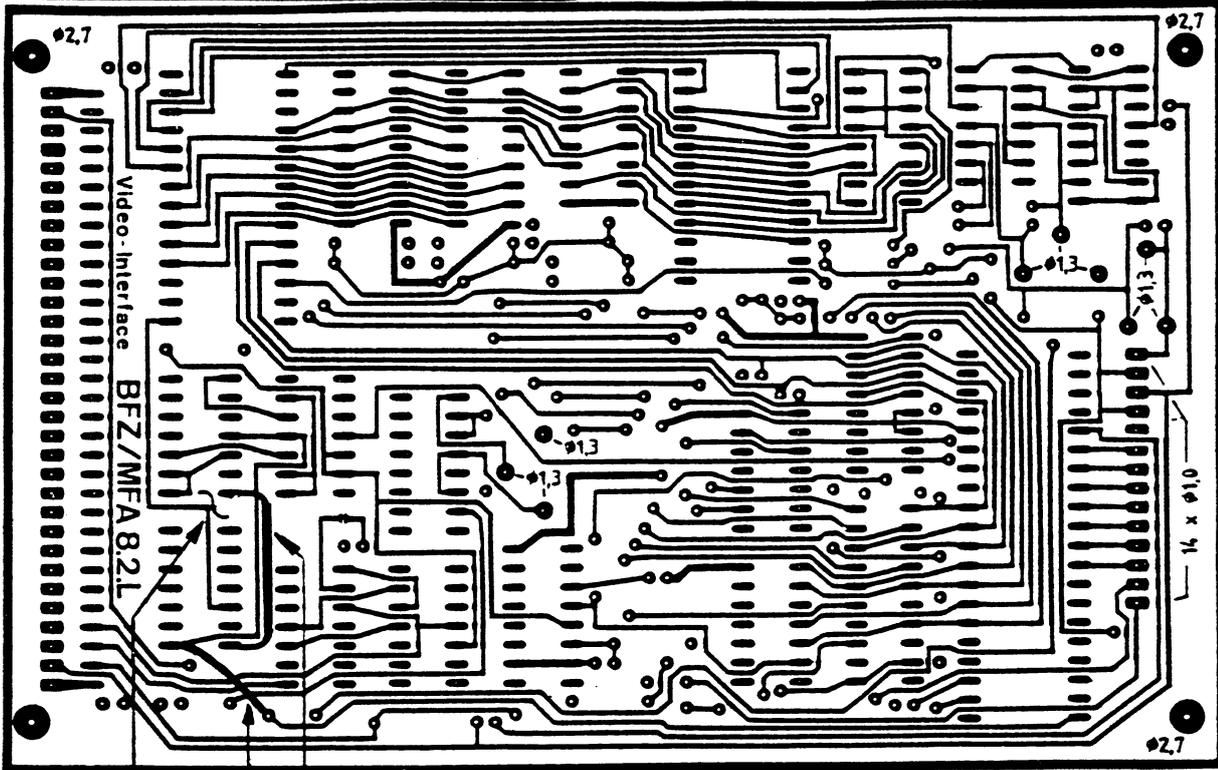
Bei der Umrüstung "MAT85 V1.8/S" und Verwendung des Video-Interface BFZ/MFA-8.2 ist bisher folgendes "Fehlerbild", das sich nicht vermeiden läßt, aufgetreten:

- Wird bei der Verwendung des MAT85-Kommandos "R_REGISTER" am Ende der Zeile (Registereinstellungen) nicht die Taste "CARRIAGE RETURN", sondern "SPACE" betätigt, so kann es passieren, daß eine weiter oben auf dem Bildschirm dargestellte Ausgabezeile unterhalb des "R_REGISTER"-Ausdrucks nochmals auf dem Bildschirm erscheint. Diese "Fehlfunktion" (Zeilenvorschub durch Zeilenüberlauf) wirkt sich nur auf die **B i l d - s c h i r m d a r s t e l l u n g** aus, jedoch nicht auf die Funktion des MFA-Rechners.

Bei der Umrüstung "MAT85 V1.8/S" und Verwendung der Video-Interface BFZ/MFA-8.4 ist bisher noch kein fehlerhaftes Verhalten aufgetreten.

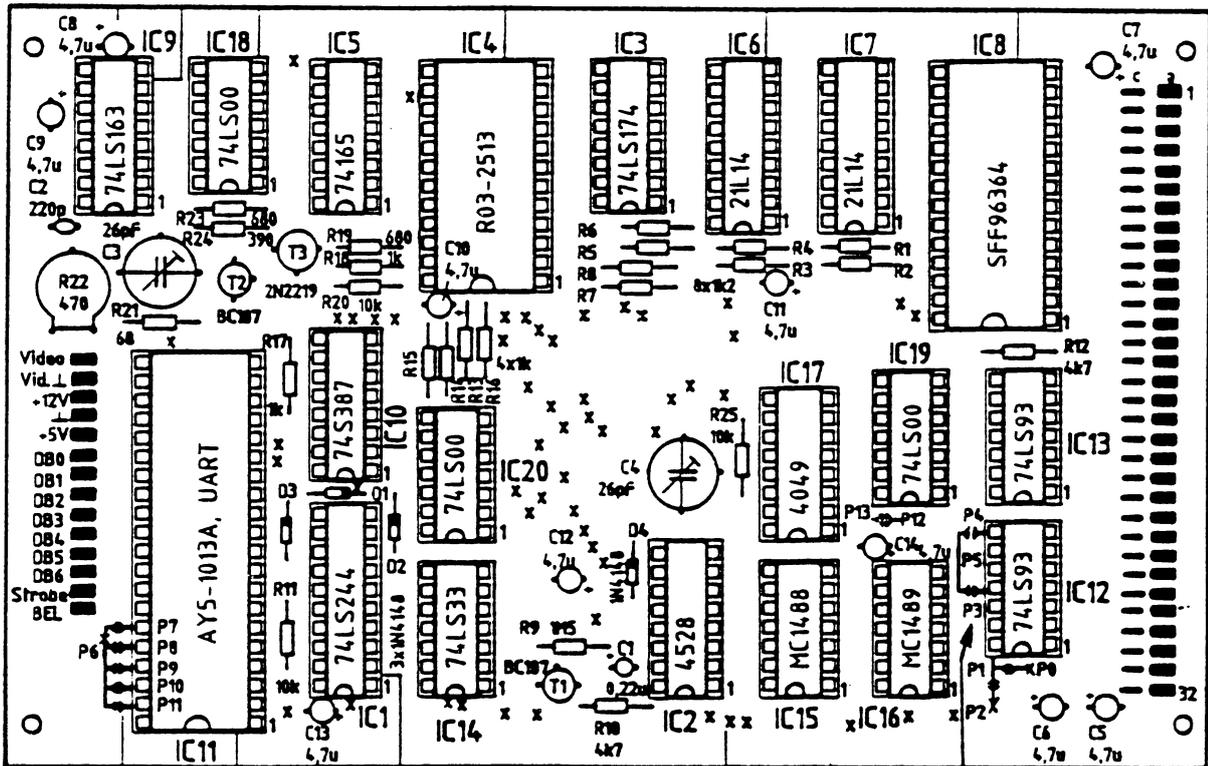
Anlage: Hardware-Änderungen auf dem Video-Interface BFZ/MFA-8.2

A) Video-Interface Leiterbahnseite



Brücke Pin 14 (IC 13) - Pin 14 (IC 12)
 Brücke Pin 1 (IC 12) - Pin 1 (IC 11)
 Abtrennen Pin 14 (IC 13)

B) Video-Interface Bestückungsseite



beide Brücken öffnen
 (P5-P3 und P5-P4)

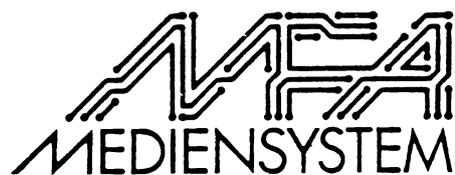
MFA-MEDIENSYSTEM

Mikrocomputer- Technik

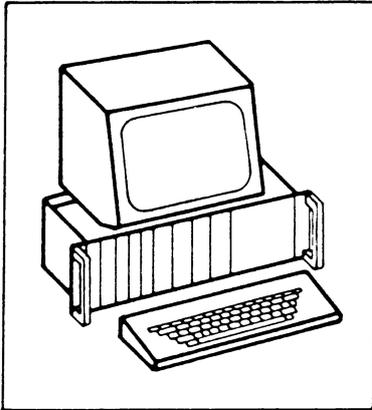
Fachpraktische Übungen · Band 4

Teil 2

Ausbaustufe PC



FACHPRAKTISCHE ÜBUNG MIKROCOMPUTER-TECHNIK



V24/RS232-
Schnittstelle

BFZ/MFA 4.10.



Inhaltsverzeichnis

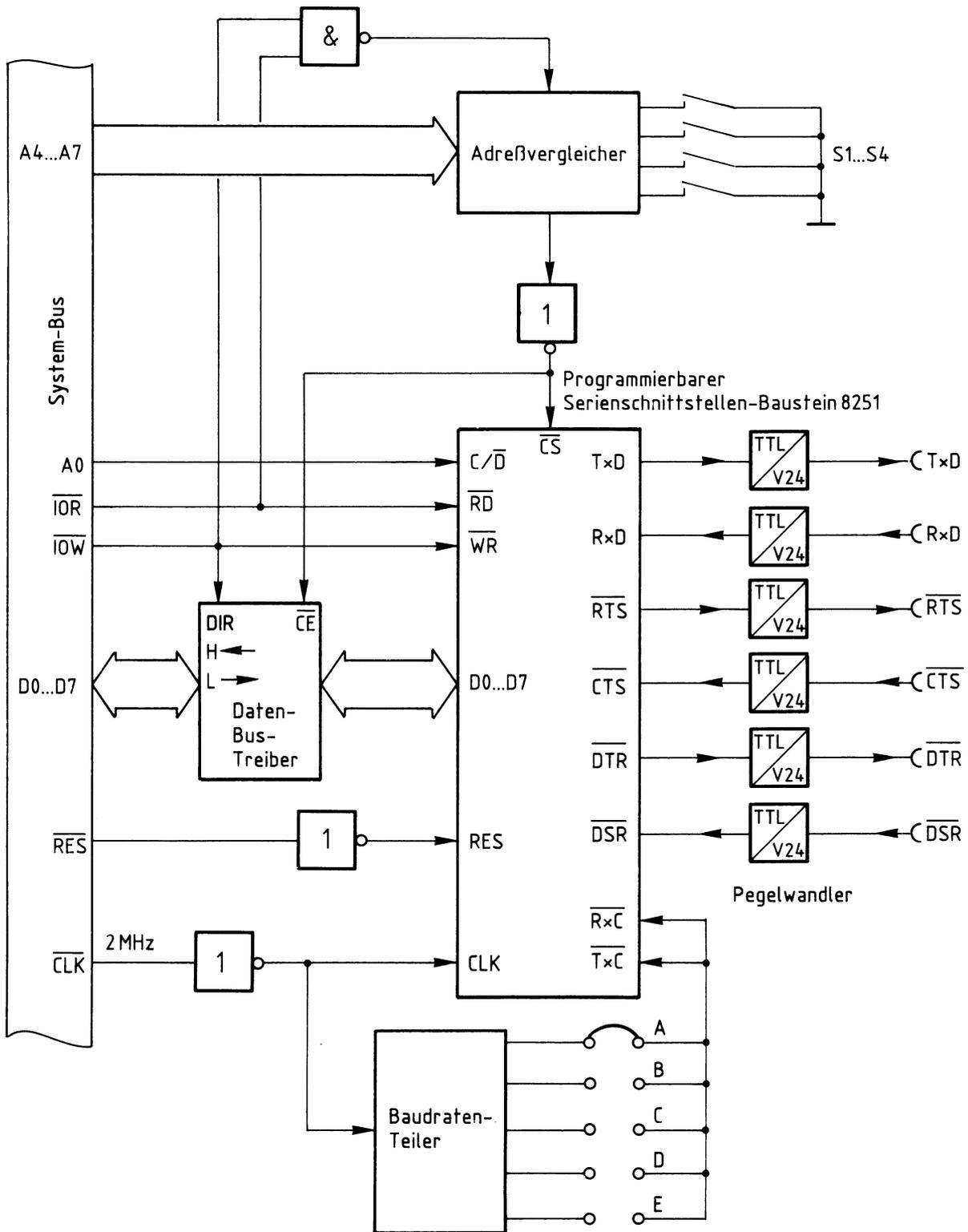
Inhalt	Seite
Hinweise	1
Blockschaltbild	2
Funktionsbeschreibung	3
Stromlaufplan	6
Bestückung Leiterplatte	7
Bereitstellungsliste zur Inbetriebnahme	9
Inbetriebnahme	10
Bereitstellungsliste	12
Bohrplan Leiterplatte	14
Layout Bestückungsseite	15
Bohrplan Frontplatte	16
Zusammenbau Baugruppe	17

Hinweise

Die Baugruppe "V24/RS232-Schnittstelle" ist eine Serienschnittstelle zur Datenübertragung zwischen dem BFZ/MFA-Computer und der Peripherie (z.B. Drucker, Datenterminals, SPS-Steuerungen). Baudraten zwischen 150 Bd und 9600 Bd sind möglich.

Die Ein- und Ausgänge besitzen ± 12 -V-Pegelwandler, deren Anschlüsse auf eine 25polige Buchse in der Frontplatte der Baugruppe geführt sind. Die Belegung der Anschlußbuchse kann durch Lötbrücken an die Erfordernisse der Peripherie angepaßt werden. Hierdurch wird der Anschluß der Geräte mit serieller Datenübertragung erleichtert.

Blockschaltbild



Funktionsbeschreibung

Die Funktionsbeschreibung bezieht sich nur auf das Blockschaltbild. Einzelheiten des verwendeten Schnittstellenbausteins 8251 werden im Band 2 der Fachpraktischen Übungen behandelt (BFZ/MFA 4.4. "Programmierbare Serienschnittstelle" und BFZ/MFA 4.4.a "Kassetten-Interface").

Die serielle Daten-Sendeleitung TxD, die serielle Daten-Empfangsleitung RxD sowie einige Steuerleitungen des Schnittstellenbausteins 8251 sind über TTL/-V24-Pegelwandler mit einer 25poligen Buchsenleiste auf der Frontplatte der Baugruppe verbunden.

Für interne Steuerzwecke erhält der Baustein am Anschluß CLK den von einem Inverter aufgefrischten 2-MHz-Systemtakt.

Aus dem System-Takt wird außerdem im Baudraten-Teiler der zur Parallel/Seriell- und Seriell/Parallel-Wandlung erforderliche Baudraten-Takt gewonnen und dem Baustein an den Anschlüssen $\overline{\text{Rx}}\overline{\text{C}}$ und $\overline{\text{Tx}}\overline{\text{C}}$ zugeführt. Durch Lötbrücken und geeignete Initialisierung können 7 verschiedene Baudraten eingestellt werden (siehe Tabelle im Stromlaufplan).

Der Daten-Bus ist über einen bidirektionalen Datenbus-Treiber in Tristate-Technik mit den Datenleitungen des Schnittstellen-Bausteins verbunden. Der Datenbus-Treiber trennt im Ruhezustand den Datenweg und gewährleistet im aktiven Zustand einen ausreichend hohen Treiberstrom in Richtung System-Bus. Zur Aktivierung des Datenbus-Treibers muß der Anschluß $\overline{\text{CE}}$ (Chip Enable) L-Pegel erhalten.

Über den Anschluß DIR (direction, Richtung) wird mit dem Signal $\overline{\text{IOW}}$ die Arbeitsrichtung des Datenbus-Treibers gesteuert. Hierdurch wird festgelegt, ob Daten vom Prozessor zum Baustein 8251 und von dort über den Anschluß TxD seriell ausgesendet werden, oder ob der Prozessor die über den Anschluß RxD seriell empfangenen Daten liest.

Die Freigabe des Datenbus-Treibers und des Schnittstellenbausteins erfolgt durch den Adreßvergleichler. Dies geschieht nur, wenn die Adresse auf den Adreßleitungen A4 bis A7 mit derjenigen Baugruppen-Nummer übereinstimmt, die mit den Schaltern S1 bis S4 eingestellt ist. Zur Freigabe des Adreßvergleichlers muß eine der Steuerleitungen $\overline{\text{IOW}}$ oder $\overline{\text{IOR}}$ aktiv sein.

Mit der Adreßleitung A0 werden bei H-Pegel die Control-Register, und bei L-Pegel die Datenregister (Sender- und Empfänger-Register) des 8151 adressiert.

Funktionsbeschreibung

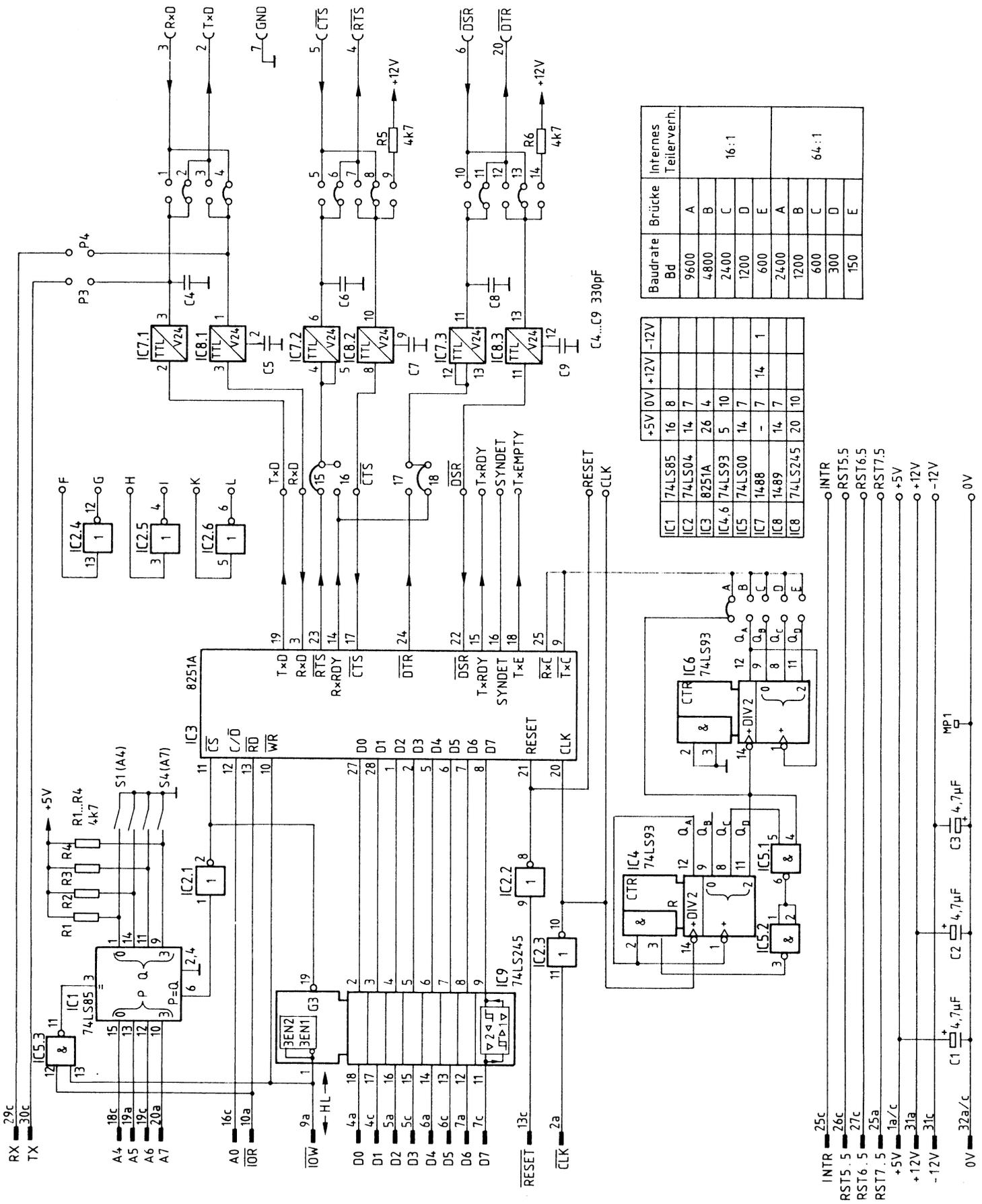
Verbindungen Schnittstellenbaustein - 25polige Buchsenleiste

- "TxD" (Transmitter-Data, Senderdaten): Serieller Datenausgang des Bausteins, durch Lötbrücke 2 mit Anschlußstift 2 der 25poligen Buchsenleiste verbunden
- "RxD" (Receiver-Data, Empfängerdaten): Serieller Dateneingang des Bausteins, durch Lötbrücke 4 mit Anschlußstift 3 der 25poligen Buchsenleiste verbunden
- " $\overline{\text{RTS}}$ " (Request To Send, Sendeteil der Peripherie einschalten): Der Signalzustand dieses beliebig verwendbaren Bausteinausgangs wird bei der Initialisierung durch das Kommandowort festgelegt und kann danach auch wieder per Programm verändert werden. In der Grundverdrahtung der V24/RS232-Schnittstelle wird das Bausteinsignal $\overline{\text{RTS}}$ über die Lötbrücke 15, den Pegelwandler IC7.2 und die Lötbrücke 6 auf den $\overline{\text{RTS}}$ -Anschluß der 25poligen Buchsenleiste geführt (Anschlußstift 4).
- "RxRDY" (Receiver Ready, Empfänger hat ein Zeichen empfangen/ist besetzt): Ausgang, der beim Vorliegen eines Zeichens im Empfängerregister des Schnittstellenbausteins H-Pegel annimmt. In der Grundverdrahtung der Schnittstelle wird RxRDY über die Brücke 18, den Pegelwandler IC7.3 und die Brücke 11 zum Anschlußstift " $\overline{\text{DTR}}$ " der 25poligen Buchsenleiste geführt. Hiermit kann der externe Datensender so lange gestoppt werden, bis das Empfängerregister des Bausteins von der CPU gelesen ("geleert") wurde und damit wieder bereit ist, ein neues Zeichen aufzunehmen.
- " $\overline{\text{CTS}}$ " (Clear To Send, Senderfreigabe): L-Signal an diesem Bausteineingang gibt den eigenen Datensender frei, H-Signal sperrt ihn. Über den Pegelwandler IC8.2 und die Lötbrücke 8 ist der $\overline{\text{CTS}}$ -Anschluß des Bausteins mit Anschlußstift 5 der 25poligen Buchse verbunden. Hier ist zur Senderfreigabe 0-Signal (+3 V bis +12 V) erforderlich. Bei 1-Signal (-3 V bis -12 V) oder offenem Eingang ist der Datensender gesperrt.
Wird anstelle der Lötbrücke 8 die Lötbrücke "9" geschlossen, so erfolgt die Senderfreigabe nicht über die Peripherie, sondern über R5 (Datensender ist ständig freigegeben).

Funktionsbeschreibung

- " $\overline{\text{DTR}}$ " (Data Terminal Ready, Betriebsbereitschaft des eigenen Gerätes): Der Signalzustand dieses beliebig verwendbaren Ausgangs wird bei der Initialisierung des Bausteins durch das Kommandowort festgelegt und kann danach auch wieder per Programm verändert werden. Mit L-Pegel am Bausteinanschluß $\overline{\text{DTR}}$ kann einem Peripheriegerät mitgeteilt werden, daß die Serienschnittstelle betriebsbereit ist.
- " $\overline{\text{DSR}}$ " (Data Set Ready, Betriebsbereitschaft eines anderen Gerätes): Der Signalzustand dieses beliebig verwendbaren Eingangs kann über das Statuswort des Schnittstellenbausteins abgefragt werden. Mit Hilfe des Anschlusses $\overline{\text{DSR}}$ ist der Prozessor hierdurch z.B. in der Lage, die Betriebsbereitschaft eines Peripheriegerätes festzustellen. Hierzu muß dieses Gerät 0-Signal (+3 V bis +12 V) auf die $\overline{\text{DSR}}$ -Leitung legen.

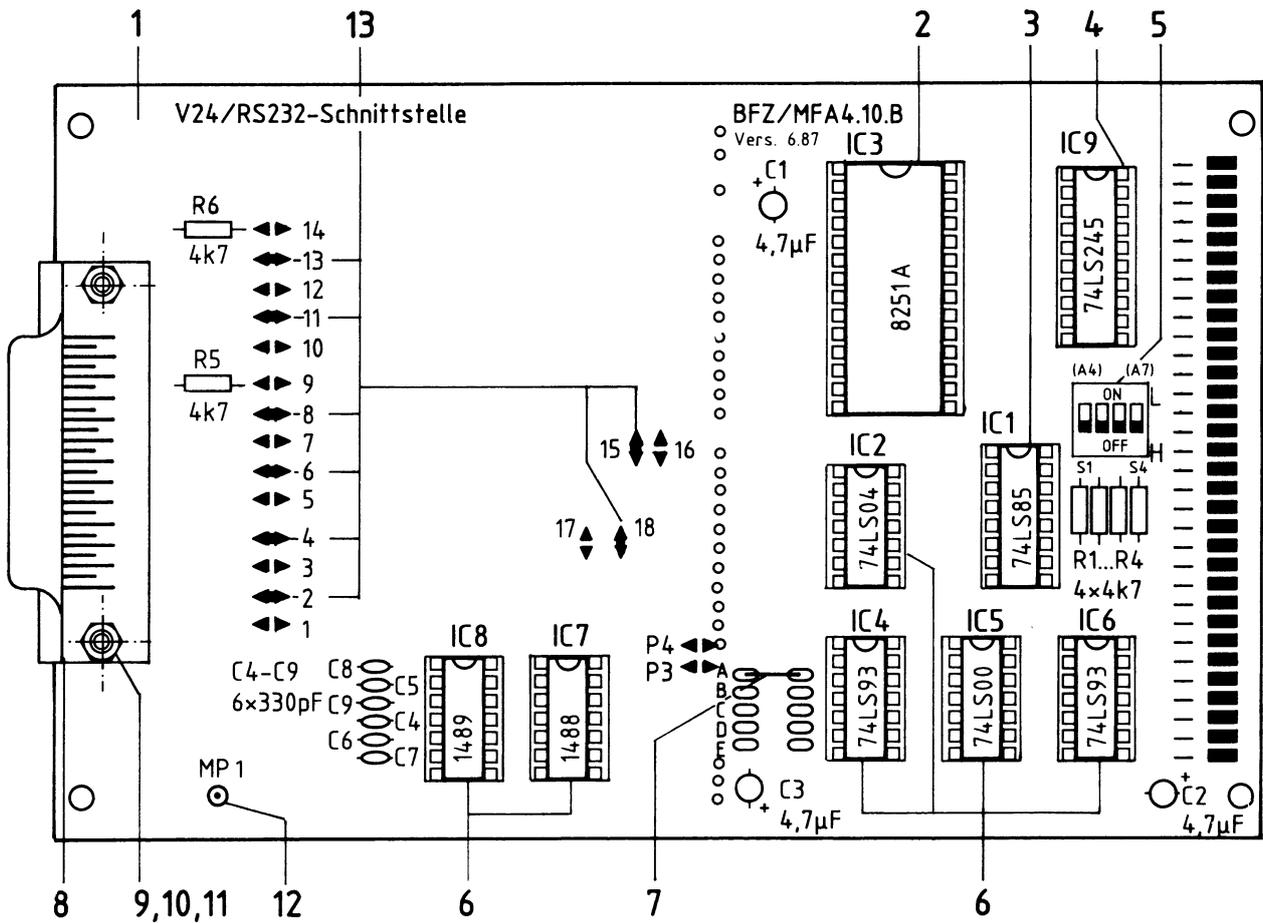
Stromlaufplan



Baudrate	Brücke	Internes Teilverh.
9600	A	
4800	B	
2400	C	16:1
1200	D	
600	E	
2400	A	
1200	B	
600	C	64:1
300	D	
150	E	

	+5V	0V	+12V	-12V
IC1	74LS85	16	8	
IC2	74LS04	14	7	
IC3	8251A	26	4	
IC4,6	74LS93	5	10	
IC5	74LS00	14	7	
IC7	1488	-	7	14
IC8	1489	14	7	
IC8	74LS245	20	10	

Bestückung Leiterplatte



Stückliste Leiterplatte BFZ/MFA 4.10.

Pos.	Stckz.	Benennung/Daten	Bemerkung
1	1	Leiterplatte BFZ/MFA 4.10.	
2	1	IC-Fassung 28polig	
3	1	IC-Fassung 16polig	
4	1	IC-Fassung 20polig	
5	1	Miniatur-Schiebeschalter 4polig	
6	6	IC-Fassung 14polig	
7	1	Lötbrücke bei "A", hergestellt aus Schaltdraht 0,5 mm	siehe auch im Stromlaufplan

Bestückung Leiterplatte

Pos.	Stckz.	Benennung/Daten	Bemerkung
8	1	Sub-D-Buchsenleiste 25polig	
9	2	Zylinderschraube M3x10 DIN 84	
10	2	Federscheibe A3,2 DIN 137	
11	2	Sechskantmutter M3 DIN 439	
12	1	Lötstift/-nagel für 1,3 mm-Loch-Ø	
13	8	Lötbrücke; bei "2", "4", "6", "8", "11", "13", "15" und "18"	Grundeinstellung der V24/RS232-Schnittst.

Bauteilliste Leiterplatte BFZ/MFA 4.10.

Kennz.	Benennung/Daten	Bemerkung
R1...R6	Widerstand 4,7 kΩ	±5%, 0,25 W
C1...C3	Tantal-Elko 4,7 µF/35 V	Tropfenform
C4...C9	Keramikkondensator 330 pF/50 V	
IC1	4-Bit-Vergleicher 74 LS 85	
IC2	Sechs Inverter 74 LS 04	
IC3	Programmierbarer Serienschnittstellen-Baustein 8251A	
IC4, IC6	4-Bit-Binärzähler 74 LS 93	
IC5	Vier NAND-Gatter 74 LS 00	
IC7	Vier Leitungstreiber für V24-Schnittstellen, MC 1488 oder 75188	
IC8	Vier Leitungsempfänger für V24-Schnittstellen, MC 1489 oder 75189	
IC9	Acht Bus-Transceiver 74 LS 245	

Bereitstellungsliste zur Inbetriebnahme

Zur Inbetriebnahme bzw. Prüfung der Baugruppe "V24/RS232-Schnittstelle" wird folgendes Material benötigt:

Stckz.	Benennung/Daten	Bemerkung	
1	Baugruppenträger mit Busverdrahtung BFZ/MFA 0.1.	Alle Baugruppen komplett aufge- baut und geprüft	
1	Bus-Abschluß BFZ/MFA 0.2.		
1	Trafo-Einschub BFZ/MFA 1.1.		
1	Spannungsregelung BFZ/MFA 1.2.		
1	Prozessor 8085 BFZ/MFA 2.1.		
1	8-K-RAM/EPROM BFZ/MFA 3.1., einge- richtet als EPROM-Baugruppe		bestückt mit MAT 85
1	8-K-RAM/EPROM BFZ/MFA 3.1., einge- richtet als RAM-Baugruppe		bestückt mit mind. zwei Stück 2-K-RAM-Bausteinen ab Adresse F800 und E000
1	ASCII-Tastatur BFZ/MFA 8.1.		
1	Video-Interface BFZ/MFA 8.2.		
1	Datensichtgerät		

Inbetriebnahme

1. Sichtkontrolle

Die Sichtkontrolle wird anhand von Stromlauf- und Bestückungsplan sowie Stück- und Bauteilliste durchgeführt.

- Sind alle Lötstellen ordnungsgemäß verlötet?
- Sind alle erforderlichen Lötbrücken geschlossen?
- Sind alle Elektrolytkondensatoren richtig gepolt?
- Sind alle ICs richtig eingesteckt?
- Sind alle Schrauben festgezogen?

2. Funktionsprüfung

Bei der Funktionsprüfung wird die V24/RS232-Schnittstelle im BFZ/MFA-Mikrocomputer betrieben und vom Betriebssystem MAT 85 initialisiert. Durch eine Drahtbrücke wird eine Verbindung zwischen dem Ausgang des seriellen Datensenders (TxD) und dem Eingang des seriellen Datenempfängers (RxD) hergestellt.

Mit Hilfe eines Prüfprogramms gibt die CPU einen Datenwert über die V24/RS232-Schnittstelle aus, empfängt diesen wieder und stellt den empfangenen Datenwert als ASCII-Zeichen auf dem Bildschirm des Datensichtgerätes dar.

2.1. Vorbereitungen zur Inbetriebnahme der V24/RS232-Schnittstelle

- Lötbrücke "9" zusätzlich geschlossen
- Alle vier Schalter zur Einstellung der Baugruppen-Nummer geöffnet (ergibt Adr. "FX")
- Verbindung zwischen TxD und RxD hergestellt (Drahtbrücke zwischen Anschluß 2 und Anschluß 3 der 25poligen Buchsenleiste)
- BFZ/MFA-Mikrocomputer mit eingesetzter V24/RS232-Schnittstelle eingeschaltet und Betriebssystem gestartet

Inbetriebnahme

2.2. Überprüfung der Schnittstelle

- Eingabe des folgenden Prüfprogramms in Assemblersprache ab Adresse F800:

```
F800  3E 2A      PRUEF: MVI A,2A    ;Akku mit Wert für das Zeichen "*"
                               ;laden
F802  CD 2108      CALL 0821    ;Unterprogramm "CASO" (Kassetten-
                               ;Out-Routine, Sendeprogramm für
                               ;serielle Schnittstelle)
F805  CD EF07      CALL 07EF    ;Unterprogramm "CASI" (Kassetten-
                               ;In-Routine, Empfangsprogramm für
                               ;serielle Schnittstelle)
F808  CD 5200      CALL 0052    ;Unterprogramm "WCHR" (Write Cha-
                               ;racter, gibt Akku-Inhalt auf dem
                               ;Bildschirm aus)
F80B  C3 00F8      JMP PRUEF    ;Zurück zum Programmanfang
F80E                      END      ;Assembler-Ende
```

- Prüfen des eingegebenen Programms mit Hilfe des Disassemblers
- Start des Programms auf Adresse F800
Wenn der Bildschirm mit dem Zeichen "*" beschrieben wird, ist die Baugruppe in Ordnung.
- Nicht vergessen: Entfernen der zusätzlich eingebauten Lötbrücke "9"

Damit ist die Inbetriebnahme beendet.

Bereitstellungsliste

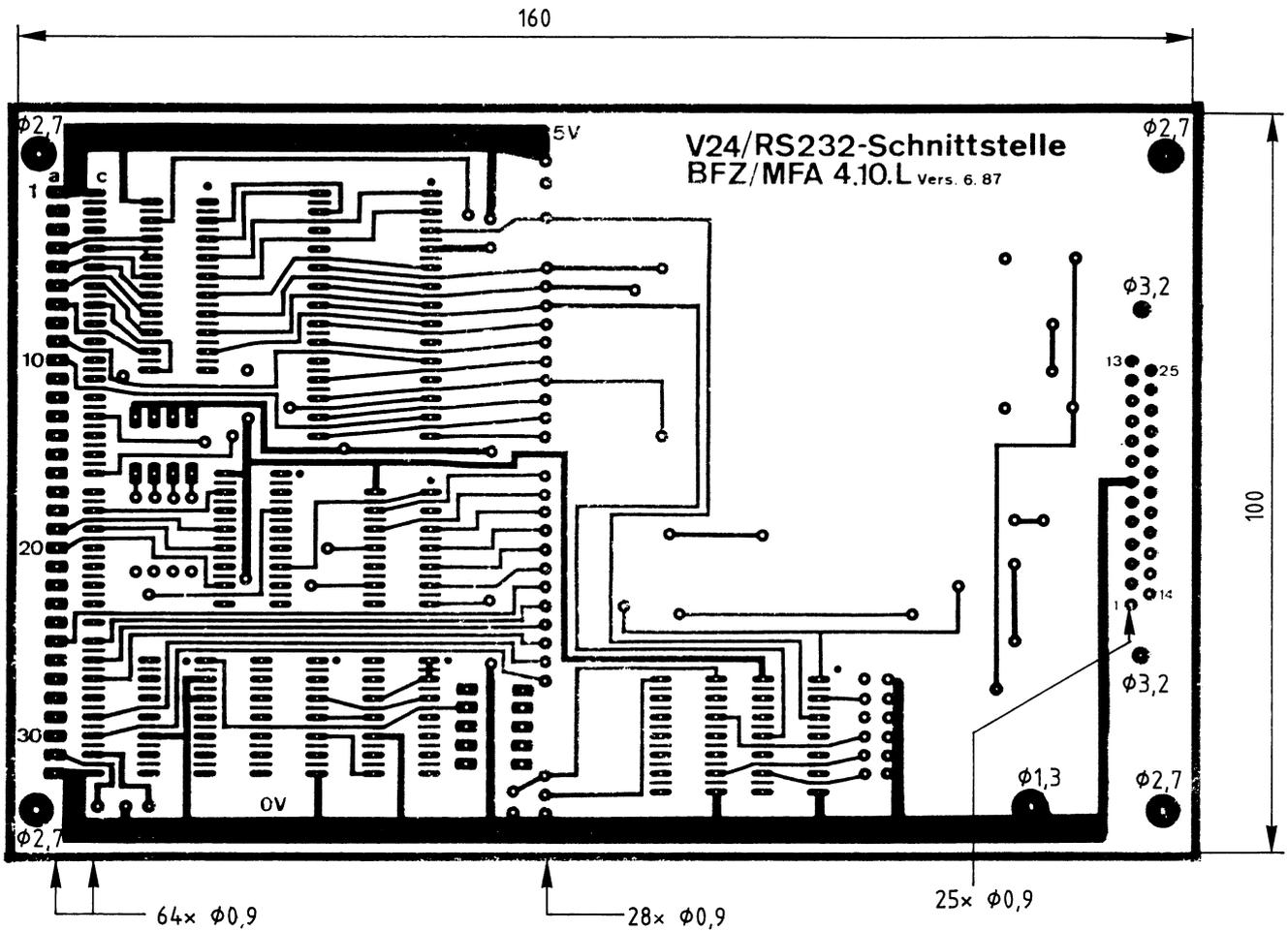
Stckz.	Benennung/Daten	Bemerkung
1	Leiterplatte, ca. 110x170 mm Mat.: Epoxid-Glashartgewebe (Hgw 2372)	doppelseitig Cu-kaschiert (35 µm) u. mit Fotolack beschichtet
je 1	Filmvorlage BFZ/MFA 4.10.L u. 4.10.B zum Belichten der Leiterplatte	je nach Ätzverfahren Pos.- oder Neg.-Film
1	Frontplatte, Teilung L-C 05 Alu, 2 mm dick, Breite: 25,1 mm	z.B. Intermas Nr. 409-017 665
1	Griff komplett mit Abdeckung T03	z.B. Intermas Nr. 409-017 927
1	Frontverbinder 1,6 FEE	z.B. Intermas Nr. 409-024 830
1	Messerleiste 64polig, DIN 41612	z.B. Erni STV-P-364 a/c Nr. 9722.333.401
2	Zylinderschraube M3x10 DIN 84	
1	Zylinderschraube M2,5x8 DIN 84	
2	Zylinderschraube M2,5x10 DIN 84	
3	Zylinderschraube M2,5x12 DIN 84	
2	Zylinderschraube mit Schaft BM2,5x10/5 DIN 84	
2	Federscheibe A3,2 DIN 137	
5	Federscheibe A2,7 DIN 137	
1	Federring B2,5 DIN 127	
2	Sechskantmutter M3 DIN 439	
4	Sechskantmutter M2,5 DIN 439	
2	Schraubensicherung, Kunststoff	z.B. Intermas Nr. 409-026 748
1	Sub-D-Buchsenleiste mit Kunststoff- winkel (mit Lochplatte vernietet), 25polig, 90° abgewinkelt zum Ein- löten in gedruckte Schaltungen	z.B. Fischer Metroplast Nr. DB WK 25
1	Miniatur-Schiebeschalter 4polig DIL	
3	Tantal-Elko 4,7 µF/35 V	Tropfenform
6	Widerstand 4,7 kΩ	±5%, 0,25 W

Bereitstellungsliste

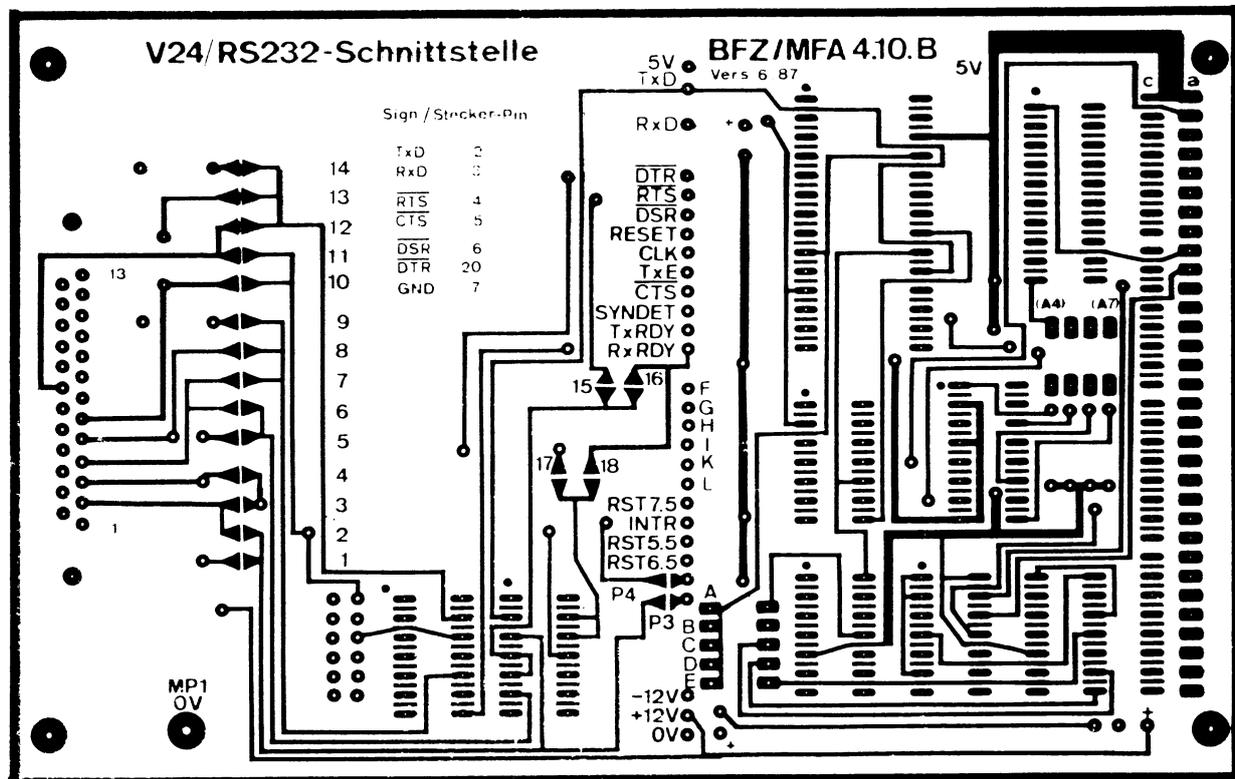
Stckz.	Benennung/Daten	Bemerkung
6	Keramikkondensator 330 pF/50 V	RM 2,5
1	IC 74 LS 04, Sechs Inverter	
1	IC 74 LS 85, 4-Bit-Vergleicher	
2	IC 74 LS 93, 4-Bit-Binärzähler	
1	IC 74 LS 00, Vier NAND-Gatter	
1	IC 74 LS 245, Acht Bus-Transceiver	
1	IC 8251A, programmierbarer serieller Schnittstellen-Baustein	
1	IC MC 1488, Vier Leitungstreiber für V24-Schnittstellen	oder SN 75188
1	IC MC 1489, Vier Leitungsempfänger für V24-Schnittstellen	oder SN 75189
6	IC-Fassung 14polig DIL	
1	IC-Fassung 16polig DIL	
1	IC-Fassung 20polig DIL	
1	IC-Fassung 28polig DIL	
1	Lötstift/Lötnagel für 1,3 mm Loch-Ø	als Meßpunkt
n.B.	Schaltdraht 0,5 mm	
n.B.	Lötendraht	
n.B.	Lötlack	
n.B.	Reinigungsmittel	zum Entfetten der Frontplatte
n.B.	Beschriftungsmaterial, Abreibe-symbole oder Tuscheschreiber	zum Beschriften der Frontplatte
n.B.	Plastik-Spray	zum Besprühen der Frontplatte

Das zur Inbetriebnahme der Baugruppe erforderliche Material ist der Bereitstellungsliste zur Inbetriebnahme zu entnehmen.

Bohrplan Leiterplatte

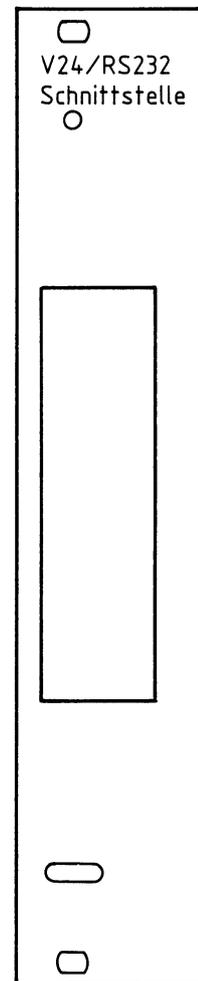
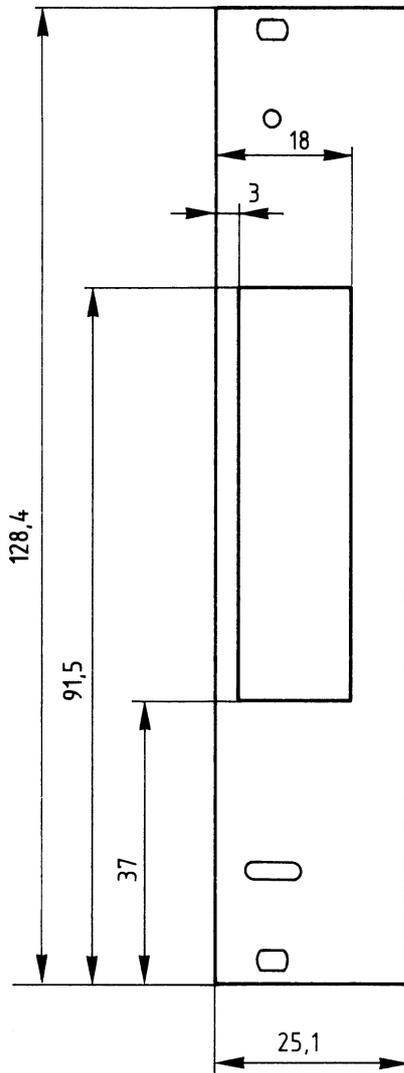


Alle nicht bemaßten Bohrungen $\phi 0,8$ mm
 Benötigte Bohrer: 0,8 - 0,9 - 1,3 - 2,7 - 3,2 mm



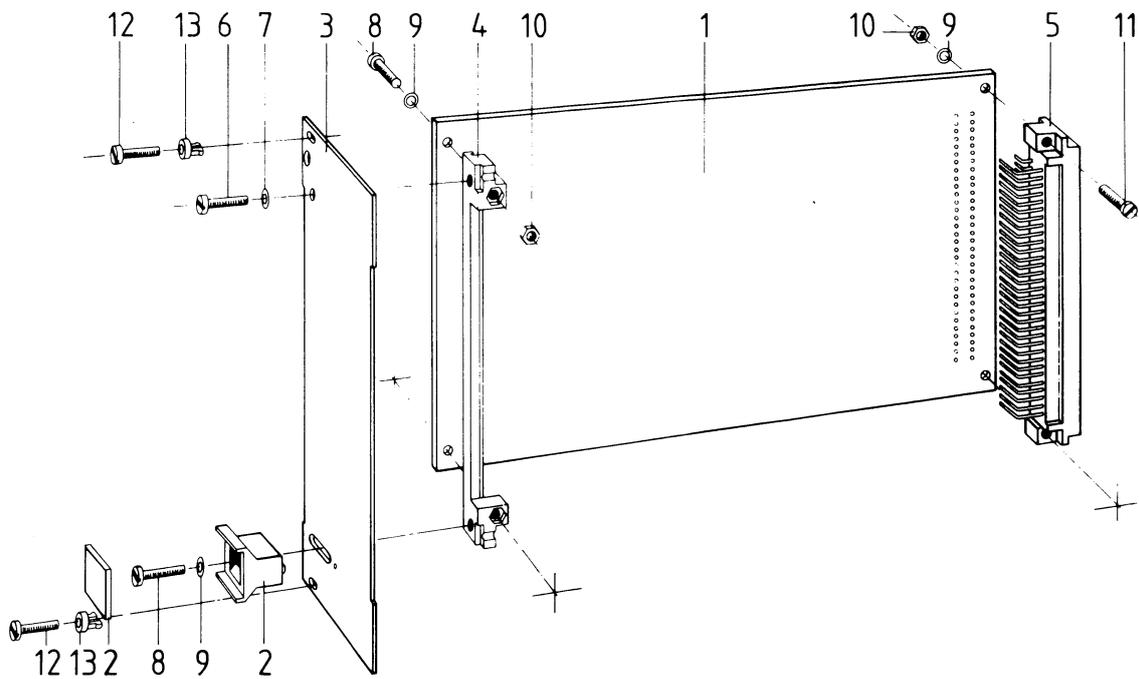
Bohrplan Frontplatte

Beschriftungsvorschlag



Material: Frontplatte L-C 05
Alu 2mm

Zusammenbau Baugruppe



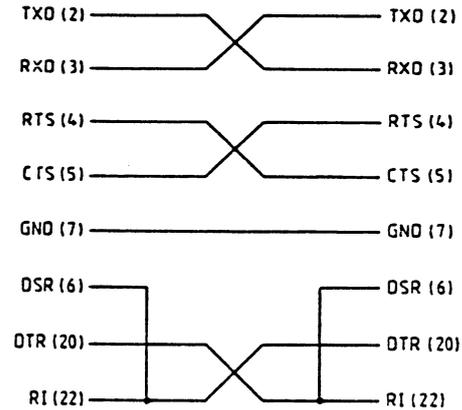
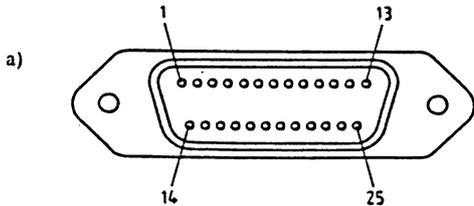
Stückliste für den Zusammenbau

Pos.	Stckz.	Benennung/Daten	Bemerkung
1	1	Leiterplatte BFZ/MFA 4.10.	komplett bestückt
2	1	Griff komplett	
3	1	Frontplatte	
4	1	Frontverbinder	
5	1	Messerleiste 64polig, DIN 41612	
6	1	Zylinderschraube M2,5x8 DIN 84	
7	1	Federring B2,5 DIN 127	
8	3	Zylinderschraube M2,5x12 DIN 84	
9	5	Federscheibe A2,7 DIN 137	
10	4	Sechskantmutter M2,5 DIN 439	
11	2	Zylinderschraube M2,5x10 DIN 84	
12	2	Zylinderschraube mit Schaft, BM2,5x10/5 DIN 84	
13	2	Schraubensicherung, Kunststoff	

Anschlußbelegung

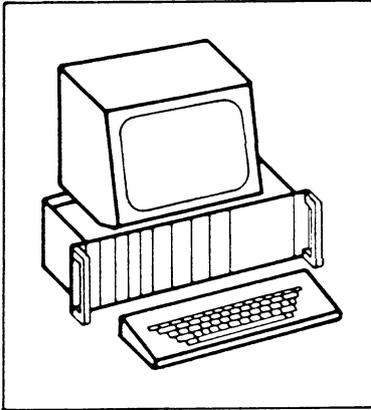
V 24/RS 232 - Datenkabel

- a) Der genormte 25polige Stecker für die V24-Schnittstelle.
 b) Tabelle der V24-Signale. Die angegebene Signalrichtung bezieht sich immer auf die Blickrichtung vom Computer bzw. Terminal zum Modem.



Stift Nr.	Signalrichtung	Bezeichnung nach			Techn. Abk.	Bezeichnung	
		DIN	V24	RS232		englisch	deutsch
1							
2	aus	D1	103	BA	TXD	Transmitted Data	Sendedaten
3	ein	D2	104	BB	RXD	Received Data	Empfangsdaten
4	aus	S2	105	CA	RTS	Request to send	Sendeteil einschalten
5	ein	M2	106	CB	CTS	Clear to send	Sendebereitschaft
6	ein	M1	107	CC	DSR	Data set ready	Betriebsbereitschaft
7		E2	102	AB	GND	Signal ground	Betriebserde
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20	ein oder aus	S1.1 S1.2	108/1 108/2	CD	DTR	Connect data set to line Data terminal ready	Übertragungsleitungen einschalten Terminal betriebs- bereit
21							
22	ein	M3	125	CE	RI	Ring indicator	Ankommender Ruf
23							
24							
25							

FACHPRAKTISCHE ÜBUNG MIKROCOMPUTER-TECHNIK



IBM-Terminalprogramm

Version 0.4

BFZ/MFA 7.4.

Das modifizierte Betriebsprogramm MAT 85 T befindet sich zusammen mit den Software-Erweiterungen (SP1, Editor, usw.) auf dem mitgelieferten 32-K-EPROM. Dieses ist auf den Boot-Platz der 64-K-RAM-Karte zu stecken; vgl. die anhängende ausführliche Beschreibung des MAT 32.



Inhaltsverzeichnis

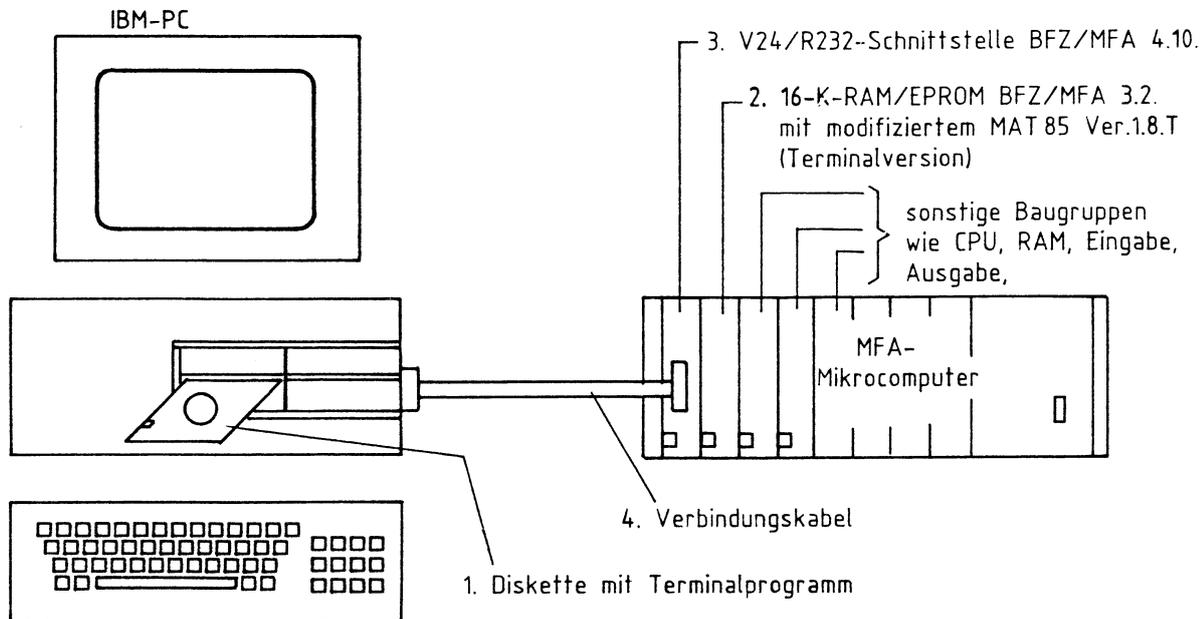
Inhalt	Seite
Hinweise	1
Gerätekonfiguration	2
Bereitstellungsliste	3
Disketten-Files	4
Betriebsprogramm MAT85	5
V24/RS232-Schnittstelle	6
Schaltsbild Serienschnittstelle	7
Bestückungsplan Serienschnittstelle	8
Verbindungskabel Serienschnittstelle	9
Start des Terminalprogramms	10 - 11
Betriebsprogramm-Kommandos für die Datensicherung	12

Hinweise

Das IBM-Terminalprogramm ermöglicht die Benutzung eines IBM-Personal-Computers (IBM-PC oder kompatibel) als Terminal zum MFA-Mikrocomputer. Damit werden das Video-Interface BFZ/MFA 8.2., die ASCII-Tastatur BFZ/MFA 8.1. sowie der externe Video-Monitor zum Aufbau der Datensicht-Station überflüssig.

Darüber hinaus ermöglicht das Terminalprogramm einen bidirektionalen Dateitransfer zwischen den beiden Computern, so daß der IBM-PC gleichzeitig als Massenspeicher für Daten und Programme aus dem MFA-Mikrocomputer verwendet werden kann. Die Aufgaben des Kassetten-Interfaces bzw. die der Floppy-Disk-Station können von diesem Terminalprogramm ebenfalls übernommen werden.

Gerätekonfiguration



Das obige Schema zeigt die Gerätekonfiguration für den Einsatz des IBM-PC als Datensicht-Station zum MFA-Mikrocomputer. Die Kommunikation zwischen den beiden Mikrocomputern erfolgt über eine V24/RS232-Verbindung mit einer voreingestellten Baud-Rate von 4800 Baud. Zur Vermeidung eines Datenverlustes wird der Datenaustausch über einen Hardware-Handshake realisiert.

An Stelle des Video-Interfaces BFZ/MFA 8.2. wird im MFA-Mikrocomputer die Baugruppe V24/RS232-Schnittstelle BFZ/MFA 4.10. verwendet, die über ein Verbindungskabel mit einer seriellen Schnittstelle (COM1 oder COM2) im IBM-PC verbunden wird.

Darüber hinaus ist im MFA-Mikrocomputer ein entsprechend angepaßtes Betriebsprogramm MAT85 Version 1.8.T für den Terminal-Betrieb über eine serielle Schnittstelle erforderlich. Dieses modifizierte Betriebsprogramm unterstützt Terminals, die kompatibel zum TeleVideo 955 sind. Das IBM-Terminalprogramm ist auf dieses geänderte Betriebsprogramm zugeschnitten.

Bereitstellungsliste

Folgende Komponenten sind für den Terminal-Betrieb über einen IBM-PC notwendig:

- 1 IBM-Terminalprogramm BFZ/MFA 7.4.
auf Diskette mit Terminaltreiber und Hilfsprogrammen
- 2 Betriebsprogramm MAT85 Version 1.8.T
für den Terminal-Betrieb mit dem TeleVideo 955 über
eine serielle Schnittstelle
- 3 V24/RS232-Schnittstelle BFZ/MFA 4.10.
- 4 Verbindungskabel
für den Anschluß des IBM-PC an den MFA-Mikro-
computer, abhängig vom verwendeten Computer PC-
seitig mit 9- bzw. 25-poliger(m) Sub-D-Buchse
oder -Stecker ausgerüstet

Folgende Komponenten werden durch den IBM-PC ersetzt bzw.
überflüssig:

Datensicht-Station bestehend aus:

- Video-Interface BFZ/MFA 8.2.
- ASCII-Tastatur BFZ/MFA 8.1.
- externer Video-Monitor

Floppy-Disk-Station bestehend aus:

- Floppy-Disk-Controller BFZ/MFA 4.7.
- 5 1/4-Zoll-Laufwerk-Station

Kassetten-Speicher bestehend aus:

- Kassetten-Interface BFZ/MFA 4.4.a
- Kassetten-Gerät

Disketten-Files

Das IBM-Terminalprogramm befindet sich auf einer IBM-kompatiblen 5 1/4-Zoll-Diskette und umfaßt die folgenden Programm- und Daten-Files:

MK.COM	Terminalprogramm
MK.MSG	Hilfstexte und Fehlermeldungen für MK.COM
MK.CFG	Konfigurationsdatei für MK.COM

Betriebsprogramm MAT85 Vers. 1.8.T

Der Anschluß einer handelsüblichen Datensicht-Station über eine V24/RS232-Schnittstelle (Baugruppen-Nummer BX) ist über ein angepaßtes Betriebsprogramm MAT85 Vers. 1.8.T für den Terminal-Betrieb möglich. In dieser Version sind im wesentlichen die Ein-/Ausgabe-Programme an die serielle Schnittstelle angepaßt. Alle anderen Programmteile sind gegenüber der ursprünglichen Version unverändert und stimmen mit dem Monitor-Listing zum MAT85 Version 1.8. überein.

Die Anpassungen, sowie einige Ergänzungen, beziehen sich auf das TeleVideo 955 - Terminal. Das IBM-Terminalprogramm ist auf diese Version zugeschnitten und reagiert auf die TeleVideo-Steuerzeichen, beispielsweise zur Positionierung des Cursors auf dem Bildschirm. Dazu werden die Steuerzeichen entsprechend vom Terminalprogramm abgefangen und umgesetzt.

Das modifizierte Betriebsprogramm befindet sich in vier EPROMs vom Typ 2716 und kann auf einer 8- bzw. 16-K-RAM/EPROM-Baugruppe verwendet werden.

Die Betriebsprogramm-Erweiterung Softwarepaket SP1 (BFZ/MFA 7.2.) kann unverändert mit dem modifizierten MAT85 zusammenarbeiten.

V24/RS232-Schnittstelle

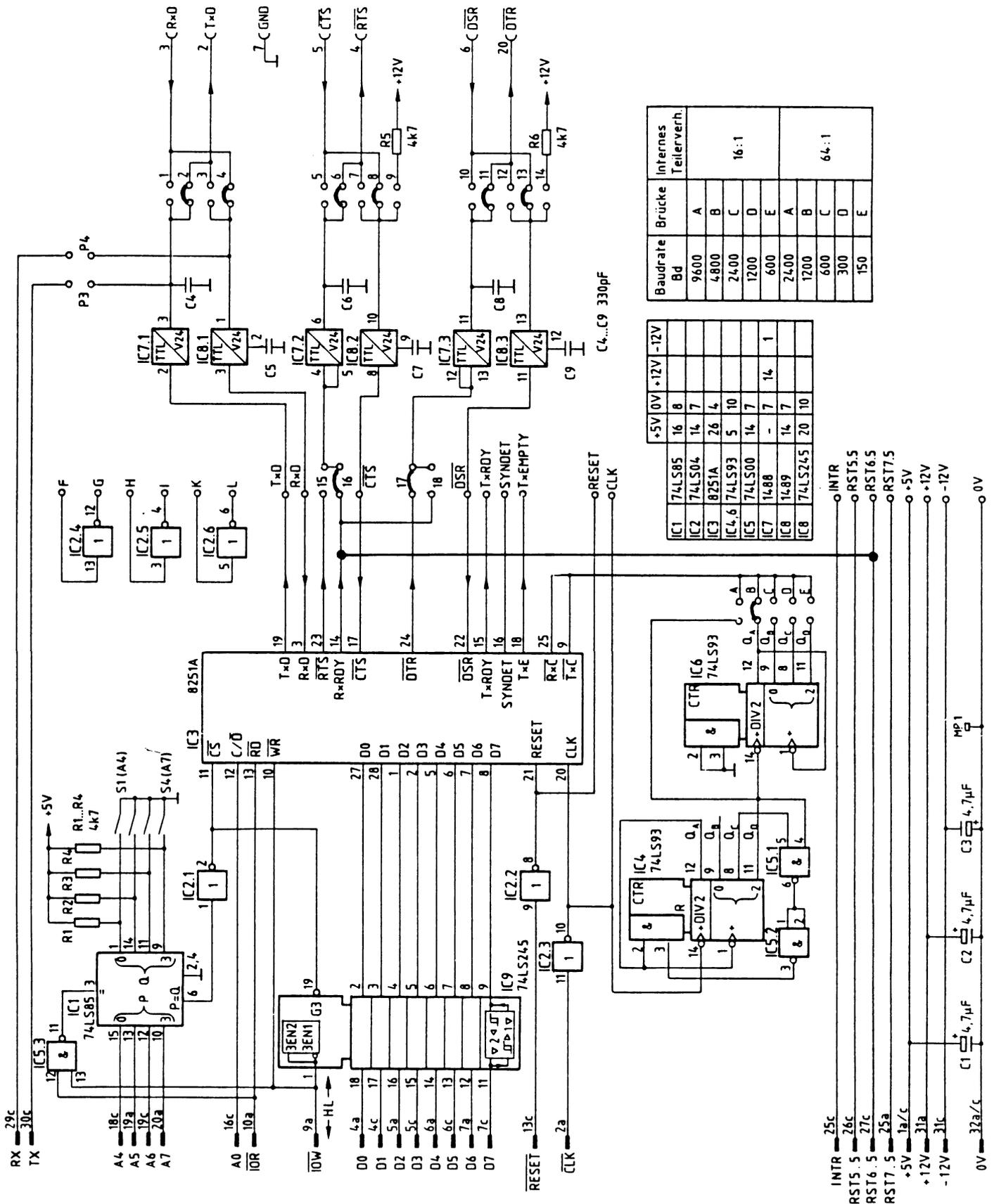
Der serielle Datenverkehr zwischen dem MFA-Mikrocomputer und dem IBM-PC erfolgt im MFA-System über eine V24/RS232-Schnittstelle BFZ/MFA 4.10. unter Ausnutzung des Hardware-Handshakes über die Steuersignale DTR bzw. RTS PC-seitig und CTS MFA-seitig. Das angepaßte Betriebsprogramm setzt bei dieser Baugruppe die Baugruppen-Nummer BX voraus.

Standardmäßig ist eine Übertragungs-Rate von 4800 Baud (Brücke B) auf der Serienschnittstelle einzustellen, die ebenfalls im IBM-Terminalprogramm voreingestellt ist. Diese läßt sich max. auf 9600 Baud erhöhen, wobei nicht sichergestellt ist, ob dann der Datenverkehr ohne Datenverlust funktioniert.

Die erforderlichen Anpassungen der Baugruppe sind im Schaltbild und dem Bestückungsplan auf den folgenden Seiten dargestellt. Zusammenfassend sind folgende Brücken, Verbindungen und Voreinstellungen vorzunehmen:

1. Baugruppen-Nummer BX einstellen
2. Übertragungs-Rate 4800 Baud einstellen (Brücke B)
3. Verbindung zwischen IC3/Pin 14 - RxRDY und der 64-poligen Steckerleiste/Pin 27c - RST 6.5 herstellen
4. Alle anderen Lötbrücken werden, wie in der Baugruppen-Beschreibung BFZ/MFA 4.10. aufgeführt, eingelötet. Dieses sind die Brücken 2,4,6,8,11, 13 16 und 17.

Schaltplan Serienschchnittstelle



Baudrate Bd	Brücke	Internes Teilerverh.
9600	A	16:1
4800	B	
2400	C	
1200	D	
600	E	64:1
2400	A	
1200	B	
600	C	
300	D	
150	E	

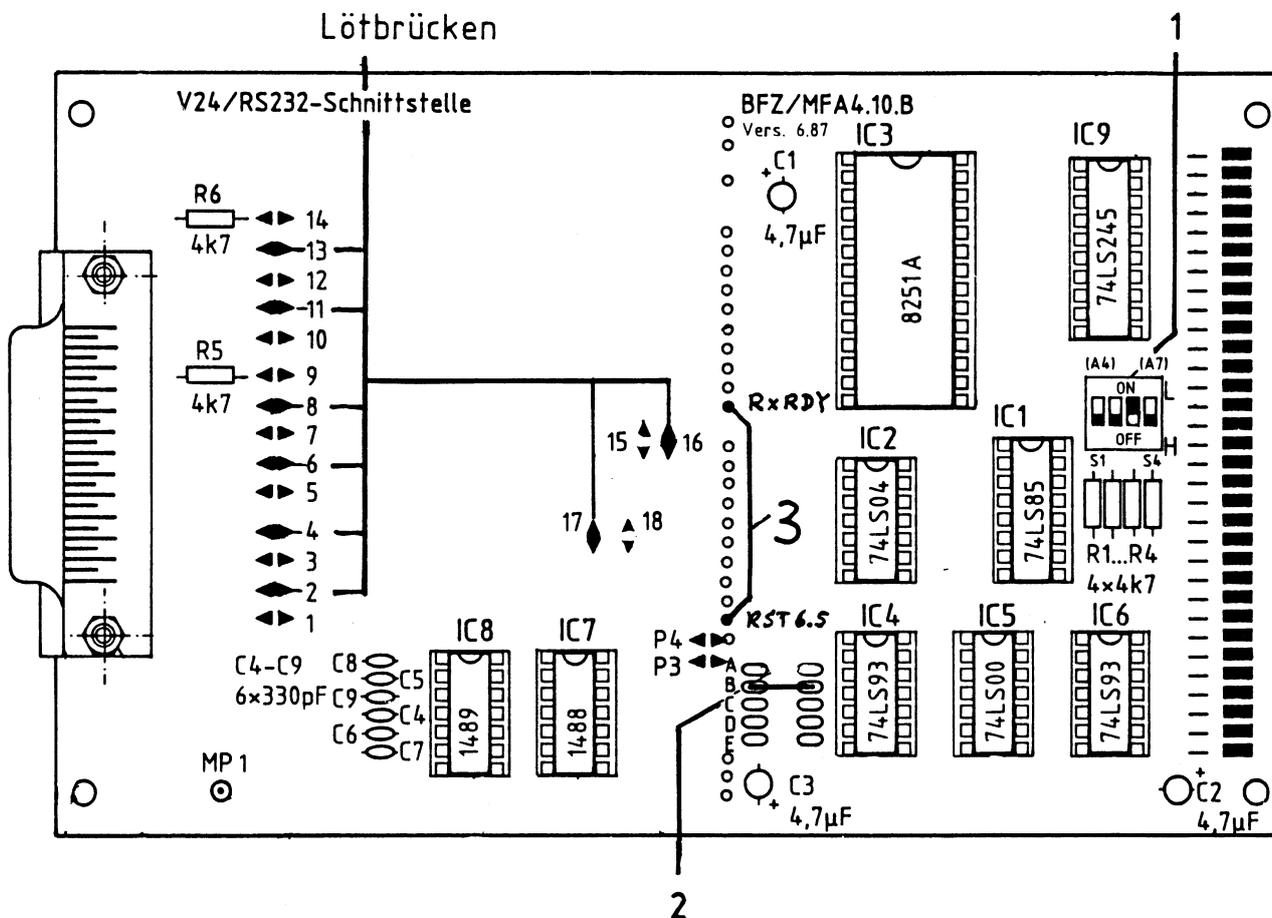
IC1	74LS85	16	8
IC2	74LS04	14	7
IC3	8251A	26	4
IC4.6	74LS93	5	10
IC5	74LS00	14	7
IC7	1488	-	7 14 1
IC8	1489	14	7
IC8	74LS245	20	10

IC1	74LS85	16	8
IC2	74LS04	14	7
IC3	8251A	26	4
IC4.6	74LS93	5	10
IC5	74LS00	14	7
IC7	1488	-	7 14 1
IC8	1489	14	7
IC8	74LS245	20	10

IC1	74LS85	16	8
IC2	74LS04	14	7
IC3	8251A	26	4
IC4.6	74LS93	5	10
IC5	74LS00	14	7
IC7	1488	-	7 14 1
IC8	1489	14	7
IC8	74LS245	20	10

IC1	74LS85	16	8
IC2	74LS04	14	7
IC3	8251A	26	4
IC4.6	74LS93	5	10
IC5	74LS00	14	7
IC7	1488	-	7 14 1
IC8	1489	14	7
IC8	74LS245	20	10

Bestückungsplan Serienschnittstelle



Anpassung für IBM-Terminalprogramm

1. Baugruppen-Nummer "BX"
2. Baudrate 4800 (Brücke B)
3. Verbindung R×RDY (Pin 14/ 8251 nach RST 6.5 (Pin 27c/ Stecker 64-polig)

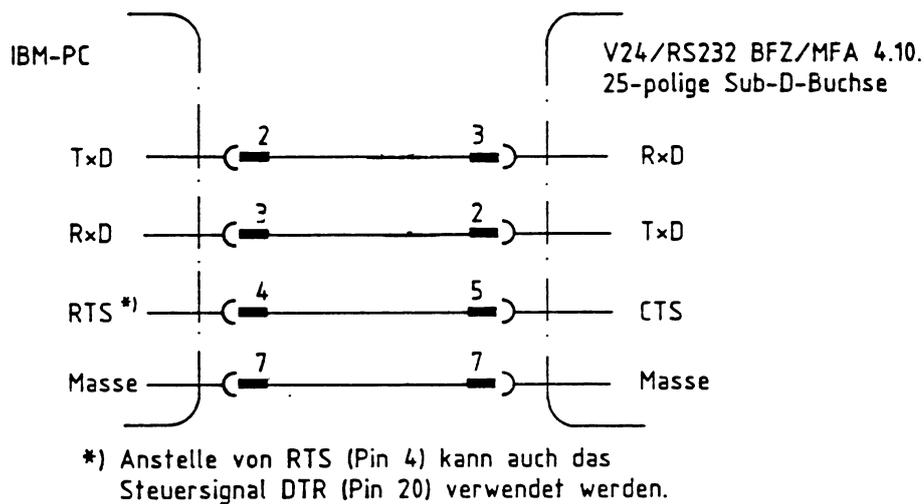
Verbindungskabel Serienschnittstelle

ACHTUNG !!!

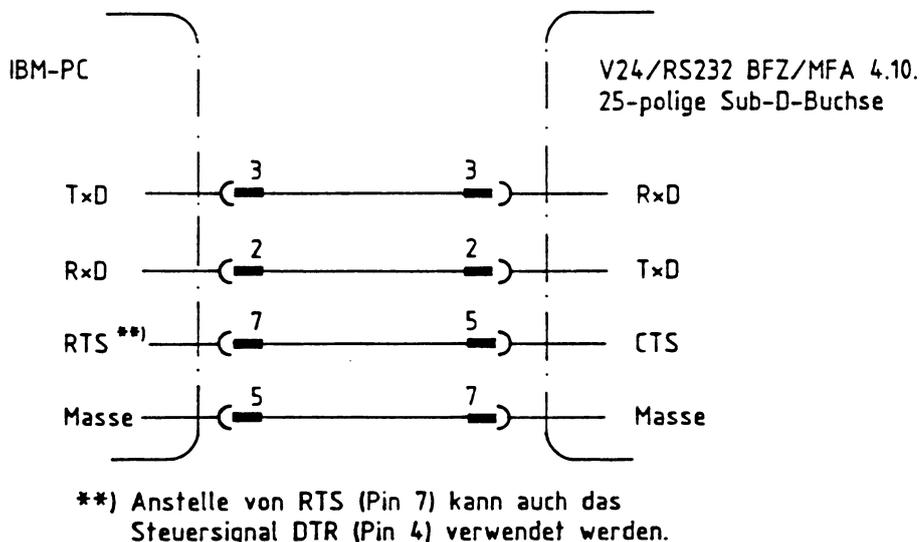
Abhängig vom verwendeten IBM-PC oder kompatiblen System können sich die Schnittstellen-Stecker PC-seitig unterscheiden. Übliche Steckverbinder sind 9- oder 25-polige Sub-D-Steckverbinder. Dabei kann es sich sowohl um Stecker wie Buchsen handeln.

Im folgenden sind die Verbindungskabel sowohl für einen 9-poligen wie für einen 25-poligen Sub-D-Stecker auf der PC-Seite dargestellt.

Version A: IBM-PC mit 25-poligem Steckverbinder



Version B: IBM-PC mit 9-poligem Steckverbinder



Start des Terminalprogramms

Das IBM-Terminalprogramm wird durch Aufruf der Programmdatei MK.COM gestartet (siehe auch Seite 4, Disketten-Files).

Nach dem Start wird zunächst ein Test der seriellen Verbindung zwischen dem MFA-System und dem IBM oder -kompatiblen PC durchgeführt. Hierbei wird die serielle Schnittstelle des PC's durch eine Software-mäßige Schleife, die ein Datentransfer zwischen TxD und RxD ausführt, überprüft. Da der Datentransfer im PC Interrupt-gesteuert abläuft, muß im PC der Interrupt-Vektor der verwendeten Serienschnittstelle (COM1 oder COM2) verändert werden. Bei fehlerhaftem Datentransfer wird der veränderte Interrupt-Vektor nicht zurückgesetzt und es empfiehlt sich ein Warmstart des PC's.

Mit jedem Neustart erfolgt automatisch der Test, ob die Verbindung zum MFA-System besteht. Im Fehlerfall erfolgt eine Meldung auf dem Monitor und der Bediener erhält die Möglichkeit, über ein Menü zwischen: erneutem Startversuch,

Abbruch des Programms und der
Anzeige von Hilfs-Texten

für den Betrieb und die Handhabung des Terminalprogramms zu wählen. Diese Hilfs-Texte können auch über eine Funktionstaste aufgerufen und sollten vor der weiteren Benutzung des Programms aufmerksam gelesen werden.

Bei einwandfreier Kommunikation zwischen PC und dem MFA-System erscheint auf dem Monitor eine Kopfzeile, in der die auf der PC-Seite benutzte Schnittstelle (COM1), die Baudrate (4800Bd) sowie die Versionsnummer des Terminalprogramms angezeigt wird. Am unteren Bildschirmrand erscheint eine Menüzeile für die Belegung der Funktionstasten, die während der Betriebsphase des Programms aktiv ist.

Auf dem Bildschirm unterhalb der Kopfzeile meldet sich das Betriebsprogramm des MFA-Mikrocomputers mit: **KMD>** und blinkendem Strichcursor (Lichtmarke).

Das System ist nun betriebsbereit.

Das MFA-Betriebsprogramm MAT85 sowie die erweiterten Funktionen unter MAT85+ (Software SP1) sind Menü-geführt, d.h. es erübrigt sich das Lernen einer Syntax (Befehle) wie es z.B. für das Betriebsprogramm MS-DOS notwendig ist.

Die Auflistung aller verfügbaren Funktionen erhält man durch Aufruf des **HELP**-Kommandos.

Ein Kommando-Aufruf erfolgt grundsätzlich durch Eingabe des ersten Buchstaben eines Kommandos und anschließender Betätigung der Taste ENTER oder RETURN (CR).

Die Funktionen unter MAT85+ (Voraussetzung ist das Vorhandensein der Erweiterung SP1) können durch Betätigung der Leertaste bzw. Space aufgerufen und auch wieder verlassen werden.

Start des Terminalprogramms

KMD> HELP

ASSEMBLER
BREAKPOINT
DISASSEMBLER
GO
HELP
IN
LOAD TAPE
MEMORY
NEXT INSTRUCTION
OUT
PRINT
REGISTER
SAVE
TRACE INTERVAL

KMD> <SPACE-Taste>

<---- Umschaltung nach MAT85+
durch Betätigen der SPACE-
Taste

KMD+> HELP

BASIC
COPY
FIND
HELP
INSERT
PROMMER
RAM-TEST
SPS
VERIFY
WRITE CONSTANT

KMD+> <SPACE-Taste>

<---- Rückkehr nach MAT85 durch
erneutes Betätigen der
SPACE-Taste

KMD>

Für die weitere Handhabung und Bedienung des Betriebsprogramms mit allen Erweiterungen muß an dieser Stelle auf folgende Literatur verwiesen werden:

- | | | |
|--|-------------|------------------|
| - MAT85 (Betriebsprogramm) | BFZ/MFA 7.1 | Fachtheoretische |
| - Software-Paket SP1
(Betriebsprogramm-Erweiterung) | BFZ/MFA 7.2 | Übungen |
| - Fachpraktische Übungen
Band 3 (Floppy-Disk-
Controller und Mini-DOS) | BFZ/MFA 4.7 | |

Betriebsprogramm-Kommandos für die Datensicherung

Folgende Kommandos sind im Betriebsprogramm und der Software-Erweiterung für die Sicherung von Daten auf dem IBM-PC vorgesehen:

- Im MAT85 wird nach Ausgabe von KMD durch Eingabe von "F" (für Floppy) in ein Menü verzweigt, das weitere Funktionen bereitstellt.
- Im Steuer-BASIC werden die beiden Kommandos "LOAD" und "SAVE" für das Laden und Sichern von BASIC-Programmen in Verbindung mit dem IBM-PC verwendet.
- Im SPS werden die beiden Kommandos "READ" und "WRITE" für das Laden und Sichern von SPS-Programmen benutzt.

Alle Programme werden als Intel-Hex-File auf dem IBM-System gespeichert und abhängig davon, ob die Sicherung aus dem MAT85, dem Steuer-BASIC oder dem SPS erfolgte, mit dem entsprechenden Datei-Bezeichner gekennzeichnet.

MFA-MEDIENSYSTEM

Mikrocomputer- Technik

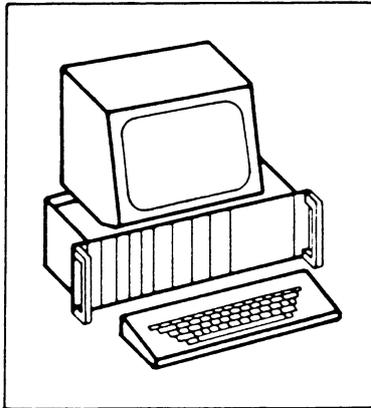
Fachpraktische Übungen · Band 4

Teil 3

Ausbaustufe MT & PC



FACHPRAKTISCHE ÜBUNG MIKROCOMPUTER-TECHNIK



MAT 32 K

bzw.

MAT 32 K-Terminalversion



Wichtiger Hinweis

Im Zuge der Weiterentwicklung des MFA-Mikrocomputers sind einige wichtige Änderungen zu beachten, die in der vorliegenden Dokumentation zum Betriebssystem MAT32 K leider noch nicht berücksichtigt werden konnten. Die hier beschriebenen Änderungen beziehen auf das MAT32 K Vers.1.8/S sowie das MAT32 K Vers.1.0/T.

Schnellere Videoausgabe für Ausbaustufe MT

Mit dem vorliegenden MAT32 K-EPROM wurde eine "schnellere Videoausgabe" durch Änderung der seriellen Ein-/Ausgabe-Routinen im Betriebsprogramm MAT85 erreicht. Die Geschwindigkeit der Zeichenübertragung zwischen Mikroprozessor und Video-Interface wurde hierbei von 1200 Bit/s auf 9600 Bit/s erhöht (Baudrate). Je nach Art der verwendeten Video-Interface-Baugruppe sind hierzu noch geringfügige Hardwareänderungen vorzunehmen. Genaue Informationen stehen in der Beilage, die Teil der vorliegenden Dokumentation des Teil 1, Ausbaustufe MT ist.

Funktionsplanausgabe für Ausbaustufe MT und PC

Außer den SPS-Erweiterungen "Kontaktplanausgabe und Dokumentation" ist in der vorliegenden MAT32 K-Version auch eine Funktionsplanausgabe implementiert. Die Funktionsplanausgabe ist ebenso wie der SPS-Interpreter an die DIN angepaßt. Der Aufruf erfolgt aus dem SPS-Interpreter durch Eingabe des Anfangsbuchstaben des Kommandonamens, hier "V" für Verknüpfungsplan ("F" für Floppy).

Speicherbelegung für Ausbaustufe MT und PC

Durch die oben beschriebene Implementierung der Funktionsplanausgabe ist der in der Dokumentation als reserviert beschriebene Speicherbereich von 6000H bis 67FFH mit dem Programm zur Funktionsplanausgabe belegt.

BFZ/MFA-Editor für Ausbaustufe MT und PC

In der Beschreibung zum BFZ/MFA-Editor ist folgendes zu beachten: Der Hinweis auf S.2 trifft nur bei Verwendung des MFA-Video-Interface BFZ/MFA-8.2 zu. Wird das Video-Interface BFZ/MFA-8.4 verwendet, ist auch in der Bildschirmdarstellung eine Unterscheidung zwischen Klein- und Großbuchstaben gegeben. Die ab S.3 beschriebene Installation des Editors auf der Grundlage verschiedener Softwarekonfigurationen stellt sich für den MAT32 K-Benutzer nicht. Sollte der Aufruf des Editors mit dem GO-Kommando erfolgen, lautet die neue Startadresse 6800H. Da das EPROM mit dem MAT32 K den Adreßbereich von 0000H-7FFFH belegt, beginnt der Edit-Buffer erst ab Adresse 8000H (oder höher) und nicht ab 6000H wie in der Beschreibung.

Wichtig für Ausbaustufe PC. Anstatt wie in der Beschreibung zur Cursor-Steuerung Control-C bitte Control-G benutzen. **Control-C** führt beim PC zum Ausstieg aus dem Terminalprogramm.

Inhaltsverzeichnis

1. Allgemeines	2
2. Beschreibung der Programmteile	2
2.1 Betriebsprogramm MAT85	3
2.2 Betriebsprogramm-Erweiterung SP1	3
2.2.1 Kommando-Erweiterung MAT85+	3
2.2.2 Steuer-BASIC-Interpreter	4
2.2.2.1 Anpassung alter BASIC-Programme	4
2.2.3 SPS-Interpreter	8
2.2.4 EPROM-Programmierer	8
2.3 Mini-DOS	8
2.4 SPS-Erweiterungen	9
2.4.1 Kontaktplanausgabe	9
2.4.2 SPS-Dokumentation	9
2.4.2.1 Programmtexter	10
2.4.2.2 Programmlader	14
2.4.2.3 Fehlerbehandlung	15
2.5 BFZ-Editor	17
3. Inbetriebnahme-Hinweise MAT32	17
Anhang	
A1 MAT85 - und MAT85+ - Kommandos	18
A2 SPS - Kommandos	19
A3 Steuer-BASIC-Befehle	20
A4 Mini-DOS - Kommandos	24
A5 EDITOR - Kommandos	25
A6 SPS - Programmbeispiel	26
A7 Speicherbelegung	29
A8 Bestückungsplan 64-KByte-RAM	30

Funktionsbeschreibung

1. Allgemeines

Das hier beschriebene Betriebsprogramm MAT32 für das MFA-Mikrocomputersystem ist, bis auf eine Erweiterung zur Dokumentation von SPS-Programmen, lediglich eine Zusammenfassung aller verfügbaren Programmteile zum Betrieb des MFA-Computers in einem 32-KByte-EPROM vom Typ 27256 und nur für den Betrieb zusammen mit der 64-KByte-RAM-Baugruppe BFZ/MFA 3.3 vorgesehen. Daher ersetzt die hier vorliegende Unterlage nicht die zu den einzelnen Programmteilen gehörenden Funktionsbeschreibungen. Für die Handhabung und Bedienung des Betriebsprogramms mit allen Erweiterungen sind daher die folgenden Fachpraktischen Übungen und Unterlagen zusätzlich erforderlich:

- | | | |
|---|-------------|---------------------------------------|
| - MAT 85 (Betriebsprogramm) | BFZ/MFA 7.1 | } in den Fach-
theoret.
Übungen |
| - Software-Paket SP 1 (Betriebs-
programm-Erweiterung) | BFZ/MFA 7.2 | |
| - Fachpraktische Übungen Band 3
(Floppy-Disk-Controller und
Mini-DOS) | BFZ/MFA 4.7 | |

In dieser Beschreibung wird daher nur auf wesentliche Unterschiede oder einzelne Erweiterungen eingegangen. Ansonsten sind die Funktionen und die Bedienungen unverändert zu den bisher ausgelieferten Programmversionen. Aus diesem Grund ist diese Beschreibung sehr kurz gefaßt.

2. Beschreibung der Programmteile

Das Betriebsprogramm MAT32 enthält folgende Programm-Module:

- Betriebsprogramm MAT85
- Betriebsprogramm-Erweiterung SP1 mit
 - ...Kommando-Erweiterung MAT85+
 - ...Steuer-BASIC-Interpreter
 - ...SPS-Interpreter
 - ...EPROM-Programmierer
- Mini-DOS
- SPS-Erweiterungen
 - ...Kontaktplanausgabe
 - ...SPS-Dokumentation mit
 Programmloader und Programmtexter
- BFZ-Editor
 - ...Hilfsmenü

Funktionsbeschreibung

Diese sind in einem 32-KByte-EPROM vom Typ 27256 zusammengefaßt. In Verbindung mit der 64-KByte-RAM-Baugruppe besteht somit die Möglichkeit, den Systemspeicher mit nur einer Speicherbaugruppe voll auszubauen. Damit stehen dem Anwender alle Systemfunktionen sowie ein 32-KByte-RAM-Speicher zur Verfügung. Der 32-KByte-EPROM-Bereich mit allen Betriebsprogrammfunktionen bleibt stets aktiv. Damit ist die gleiche Betriebssicherheit wie bei der Nutzung der bisher verwendeten EPROM-Baugruppen gewährleistet.

Diese Konfiguration kann von Vorteil sein, wenn eine Reihe von Steckplätzen im Baugruppenträger beispielsweise für Interface-Baugruppen im Rahmen einer speziellen Anwendungen benötigt werden.

Es wird aber empfohlen, diese Ausbaustufe des MFA-Mikrocomputers lediglich im fortgeschrittenen Stadium der MC-Ausbildung oder im Rahmen der Unterrichtsvorbereitung zu nutzen und die didaktischen Vorteile der stufenweisen Ausbaubarkeit des Systems in der Vermittlung von MC-Kenntnissen für die Auszubildenden und Schüler nicht zu übersehen.

2.1 Betriebsprogramm MAT85

Das Betriebsprogramm MAT85 ist nur soweit verändert worden, daß die Baud-Raten-Erfassung nach dem Einschalten der Betriebsspannung zusammen mit der Betätigung der SPACE-Taste wegfällt. Daher meldet sich das Betriebssystem unmittelbar nach dem Einschalten mit dem Ausdruck der MAT85-Kommandos.

Die Zeitkonstante für die Baud-Rate der serielle Datenübertragung zum Video-Interface, die normal mit der Abfrage der SPACE-Taste ermittelt wird, wird nach einem Kaltstart des Gerätes direkt im RAM-Speicher unter der Adresse FCDFH und FCEOH (STIME) abgelegt.

2.2 Betriebsprogramm-Erweiterung SP1

Die Betriebsprogramm-Erweiterung SP1 besteht aus den vier Modulen Kommando-Erweiterung MAT85+, Steuer-BASIC-Interpreter, SPS-Interpreter und EPROM-Programmierer. Im wesentlichen wurden zwei Änderungen vorgenommen. Die erste Änderung bezieht sich auf das Steuer-BASIC. Um den Betrieb mit einem 32-KByte-EPROM im Systemspeicher zu ermöglichen, mußte der Arbeitsspeicher des BASIC-Interpreters von der Adresse 6000H nach 80C0H verlegt werden. Die zweite Änderung bezieht sich auf den SPS-Interpreter, der an die Programmierung mit den DIN-Symbolen angepaßt wurde.

2.2.1 Kommando-Erweiterung MAT85+

In diesem Modul wurden keine Änderungen vorgenommen.

Funktionsbeschreibung

2.2.2 Steuer-BASIC-Interpreter

Der Steuer-BASIC-Interpreter wurde so verändert, daß der Arbeitsspeicher ab der Adresse 8000H, d.h. im zweiten 32-KByte-Block des Systemspeichers liegt. Hier beginnt bei Verwendung eines 32-KByte-EPROMs vom Typ 27256 mit der Betriebssoftware der RAM-Bereich auf der 64-KByte-Speicherbaugruppe. Der maximal zur Verfügung stehende Arbeitsspeicher ist somit gegenüber der normalen RAM-Startadresse von 6000H um 8-KByte kleiner geworden. Dies dürfte kein Problem darstellen, da das Steuer-BASIC nicht für extrem große Programme vorgesehen ist.

2.2.2.1 Anpassung alter BASIC-Programme

BASIC-Programme, die mittels der normalen Betriebsprogrammversion für die RAM-Startadresse 6000H entwickelt wurden und auf einer Diskette gespeichert sind, können an den neuen Arbeitsspeicher angepaßt werden. Die Schritte werden im folgenden beschrieben. Dazu sei das folgende Programmbeispiel betrachtet, welches unter der Normalversion für den RAM-Speicher ab 6000H erstellt wurde.

```
KMD+> BASIC
BFZ-STEUER-BASIC V2.4

READY
>FOR I=1 TO 10
>20 PRINT I
>30 NEXT I
>QUIT
```

Betrachtet man nach der Programmeingabe den BASIC-Arbeitsspeicher so erkennt man, daß das Programm ab der Adresse 606FH im ASCII-Code im Speicher abgelegt ist. Die einzelnen BASIC-Programmzeilen beginnen allerdings mit der jeweiligen Zeilennummer im Hexadezimal-Code. Die Zeilennummer "10" (hexadezimal 000A) der ersten Programmzeile ist in den Speicherstellen 606FH und 6070H abgelegt. Im folgenden ist der Ausdruck des BASIC-Arbeitsspeichers von 6060H bis 609FH sowohl im Hexadezimal- als auch im ASCII-Code abgebildet. Das letzte Zeichen im Programmtext ist ein CR-Steuerzeichen (0DH).

Funktionsbeschreibung

```

KMD > PRINT
START-ADR =0000 6060
STOP -ADR =0000 609F
FORMAT    =H          +----->+ Zeiger auf die nächste
                        !       ! freie Speicherstelle im
6060 00 00 00 00 00 92 60 4E DF      ! Arbeitsspeicher
6068 00 00 AF DF 00 E0 0D 0A      !
6070 00 46 4F 52 20 49 3D 31      !
6078 20 54 4F 20 31 30 0D 14      !
6080 00 50 52 49 4E 54 20 49      !
6088 0D 1E 00 4E 45 58 54 20      !
6090 49 0D>00 00 00 00 00 00      !
6098 00 00!00 00 00 00 00 00      !
                        !       !
                        +<-----+

```

```

KMD > PRINT
START-ADR =6060
STOP -ADR =609F
FORMAT    =H A

6060 00 00 00 00 00 92 .' .N DF
6068 00 00 AF DF 00 E0 0D 0A
6070 00 .F .O .R . .I .= .1
6078 . .T .O . .1 .0 0D 14
6080 00 .P .R .I .N .T . .I
6088 0D 1E 00 .N .E .X .T .
6090 .I 0D 00 00 00 00 00 00
6098 00 00 00 00 00 00 00

```

In den Speicherstellen 6064H und 6065H verwaltet der BASIC-Interpreter eine Adresse, die auf die nächste freie Speicherstelle im Arbeitsspeicher zeigt. In diesem Beispiel ist dies die Adresse 6092H. Beachten Sie, daß Adressen mit dem Low-Byte beginnend im Speicher abgelegt werden. Diese Adresse ist im neuen Arbeitsspeicher mit der Startadresse 8000H zu korrigieren.

Wird dieses Programm über den Aufruf der Floppy-Funktionen aus dem Steuer-BASIC heraus auf eine Diskette gespeichert, so legt das MiniDOS für den Programmtext eine Datei auf der Diskette an, in der auch die Adressen des Textanfangs und des Textendes gespeichert werden. Beim Zurückladen eines BASIC-Programms von der Diskette wird insbesondere die Zeigeradresse unter 6064H und 6065H auf die nächste freie Speicherstelle im BASIC-Arbeitsspeicher korrigiert.

Soll nun ein BASIC-Programm, welches im Arbeitsspeicher ab 6000H erstellt und auf der Diskette gespeichert wurde, an den neuen Arbeitsspeicher angepaßt werden, so kann das Programm nicht über den Aufruf der FLOPPY-Funktionen aus dem BASIC-Interpreter geladen werden. Folgende Schritte sind auszuführen:

Funktionsbeschreibung

1. Schritt

Start des BASIC-Interpreters und sofortiges verlassen über QUIT.

```
KMD+> BASIC
BFZ-STEUER-BASIC V2.4
```

```
READY
>QUIT
```

2. Schritt

Aufruf der Floppy-Funktionen aus dem MAT85 nach Rückkehr aus dem BASIC und laden des BASIC-Programms von der Diskette durch Eingabe des Programmnamens ohne die Erweiterung ".BAS". Anstelle der Vorschlagsadresse 606FH für das Zurückladen des BASIC-Programms wird hier die Adresse 806FH eingegeben.

```
KMD> F
BFZ-MINI-DOS V1.4
```

```
...
MENUE:
```

```
  DIRECTORY
  ERASE
```

```
...
...

```

```
BITTE BUCHSTABE EINGEBEN: LOAD
```

```
...
NAME: TEST
```

```
  START-ADRESSE: 606F 806F <----- neue Zieladresse für den
  ...                ---- BASIC-Arbeitsspeicher ab
  ...                8000H
```

```
BITTE BUCHSTABE EINGEBEN: QUIT
```

3. Schritt

Mittels des PRINT-Kommandos kann der Programmtext im BASIC-Arbeitsspeicher kontrolliert werden. Dabei erkennt man, daß der Zeiger für das Programmende auf die Speicherstelle 806FH zeigt. Das bedeutet für den BASIC-Interpreter, daß der Programmspeicher leer ist. Dies ist richtig, weil im ersten Schritt nach dem Aufruf des BASIC-Interpreters kein Programm eingegeben wurde. Diese Adresse muß mit Hilfe des MEMORY-Kommandos korrigiert werden. Dazu wird zunächst das Ende des Programmtextes ermittelt. Der folgende Speicherausdruck zeigt, daß die nächste freie Speicherstelle hinter dem Programmtext bei der Adresse 8092H zu finden ist.

 Funktionsbeschreibung

```

KMD > PRINT
START-ADR =0000 8060
STOP -ADR =0000 809F
FORMAT    =H          +----->+
                !_____!
8060 00 00 00 00 00 6F 80 4E DF      !
8068 00 00 AF DF 00 E0 0D 0A <---+ zeigt auf den Anfang des
8070 00 46 4F 52 20 49 3D 31        Programmspeichers, das be-
8078 20 54 4F 20 31 30 0D 14        deutet, der Programmspei-
8080 00 50 52 49 4E 54 20 49        cher ist leer
8088 0D 1E 00 4E 45 58 54 20
8090 49 0D>00 00 00 00 00 00
8098 00 00!00 00 00 00 00 00
                !
                +<----- nächste freie Speicherstelle
                        hinter dem Programmtext ist 8092H
  
```

Bei langen BASIC-Programmen kann das Programmende mit Hilfe des FIND-Kommandos aus dem MAT85+ durch Aufsuchen einer Folge von 00-Bytes ab dem Speicherbereich 806FH gefunden werden.

4. Schritt

Die Adresse der ersten freien Speicherstelle hinter dem Programmtext muß unter den Adressen 8064H und 8065H mit dem Low-Byte beginnend abgelegt werden. Zur Sicherheit empfiehlt sich eine Kontrolle über das PRINT-Kommando.

```

KMD> MEMORY
START-ADR =0000 8064
FORMAT    =H
8064 6F 92      <----- Low-Byte
8065 80        <----- High-Byte (hier unverändert)
  
```

```

KMD > PRINT
START-ADR =8060
STOP -ADR =809F
FORMAT    =H          +----->+ korrigierter Zeiger auf
                !_____! die nächste freie Spei-
8060 00 00 00 00 00 92 80 4E DF      ! cherstelle im Arbeits-
8068 00 00 AF DF 00 E0 0D 0A        ! speicher
8078 20 54 4F 20 31 30 0D 14        !
8080 00 50 52 49 4E 54 20 49        !
8088 0D 1E 00 4E 45 58 54 20        !
8090 49 0D>00 00 00 00 00 00        !
8098 00 00!00 00 00 00 00 00        !
                !
                +<----->+
  
```

Funktionsbeschreibung

5. Schritt

Wird nun der BASIC-Interpreter erneut aufgerufen, so meldet sich dieser mit einem RESTART. Nach Eingabe des LIST-Kommandos wird das BASIC-Programm auf dem Bildschirm ausgegeben und kann nun wieder zur Sicherung auf einer Diskette gespeichert werden.

```
KMD+> BASIC
BFZ-STEUER-BASIC V2.4 RESTART
```

```
READY
>LIST
10 FOR I=1 TO 10
20 PRINT I
30 NEXT I
```

```
READY
```

2.2.3 SPS-Interpreter

Im SPS-Interpreter wurden lediglich die Symbole für die UND- und ODER-Verknüpfung sowie für die Signalumkehr (NEGATION) an die DIN-Programmierung angepaßt. In der folgenden Tabelle sind die geänderten Zeichen zusammengefaßt:

! Operation	! altes Symbol	! neues Symbol (DIN)
! UND	! *	! U
! ODER	! +	! O
! NICHT	! /	! N

Die SPS-Erweiterungen zur Kontaktplanausgabe und zur Dokumentation von SPS-Programmen mit dem BFZ-Editor werden im Kapitel 2.4 beschrieben.

2.2.4 EPROM-Programmierer

In diesem Modul wurden keine Änderungen vorgenommen.

2.3 Mini-DOS

Das Mini-DOS zum Betrieb einer Floppy-Disk-Station wurde funktionsmäßig nicht verändert. Eine Anpassung erfolgte lediglich im Zusammenhang mit der Speicherung von BASIC-Programmen auf Diskette durch den geänderten BASIC-Arbeitsspeicher ab der Adresse 8000H.

Funktionsbeschreibung

2.4 SPS-Erweiterungen

In diesem 32-KByte-EPROM sind zwei SPS-Erweiterungen implementiert, die Kontakplanausgabe und eine Erweiterung zur Erstellung und Dokumentation von SPS-Programmen mit dem BFZ-Editor.

2.4.1 Kontaktplanausgabe

Die Kontaktplanausgabe ist ebenso wie der SPS-Interpreter an die DIN-Programmierung angepaßt. Der Aufruf erfolgt aus dem SPS-Interpreter heraus durch Eingabe des Anfangsbuchstabens des Kommandonamens, hier "K". Dieser Kommandoname wird mit dem Aufruf der HELP-Funktion im SPS nicht aufgelistet.

Die Kontakplanausgabe erstellt aus der Anweisungsliste eines SPS-Programms einen sogenannten Kontaktplan, der in seiner Symbolik an die Darstellung von Schützsaltungen angenähert ist. Im wesentlichen werden folgende Symbole verwendet:

FUNKTION	SPS-ANWEISUNG	KONTAKTPLANDARSTELLUNG
Teste auf H-Signal	z.B. UE00	-I I- Schließer *)
Teste auf L-Signal	z.B. ONM03	-I/I- Öffner *)
Ausgang auf H	z.B. =A12	-()- Schütz
Setze Ausgang auf H	z.B. =SA00	-(S)- Schütz mit A00 Selbsthaltung

*) Anstelle der eckigen Klammern für die Kontakte wird in dieser Beschreibung das Zeichen "I" verwendet. Die eckigen Klammern werden sonst auf einem Drucker, der auf den deutschen ASCII-Zeichensatz eingestellt ist, als Umlaute gedruckt.

Alle anderen Möglichkeiten werden ähnlich dargestellt. Handelt es sich in der Anweisungsliste um UND-Verknüpfungen, so wird eine Reihenschaltung der Kontakte erzeugt. Jede ODER-Verknüpfung erzeugt dagegen einen neuen parallelen Strompfad.

2.4.2 SPS-Dokumentation

Die Software-Erweiterung zur Dokumentation von SPS-Programmen ermöglicht die Erstellung bzw. Kommentierung von SPS-Programmen in Verbindung mit dem BFZ-Editor (siehe Abschnitt 2.5).

Dabei können einerseits mit dem SPS-Interpreter erstellte Programme aus dem SPS-Arbeitsspeicher in den Textpuffer des Editors übertragen und beispielsweise mit Kommentar versehen werden, damit die Lesbarkeit der Programme verbessert wird. Andererseits können kommentierte SPS-Programme direkt mit dem Editor erstellt

Funktionsbeschreibung

werden. Danach kann das SPS-Programm aus dem Textpuffer des Editors in den SPS-Arbeitsspeicher geladen werden, wobei alle Kommentare und Trennzeichen bei der Übertragung unterdrückt werden.

Die Übertragung von SPS-Anweisungen aus dem SPS-Arbeitsspeicher in den Textpuffer des Editors erfolgt über den "Programmtexter". Das Laden von SPS-Anweisungen aus dem Textpuffer des Editors in den SPS-Arbeitsspeicher wird von dem "Programmlader" vorgenommen.

! SPS-Erweiterung !	Übertragungsrichtung !
! Programmtexter !	SPS-Arbeitsspeicher --> Editor-Textpuffer !
! Programmlader !	Editor-Textpuffer --> SPS-Arbeitsspeicher !

2.4.2.1 Programmtexter

Der Programmtexter übernimmt die Übertragung von SPS-Programmen in den Textpuffer des Editors. Dabei fügt er zwischen den Operationssymbolen (z.B. U, ON, =) und den Operanden (z.B. E, A, M) Leerzeichen zur besseren Lesbarkeit ein. Ebenso trennt er den Operanden von der Operandenadresse. Darüber hinaus ergänzt er jede Anweisung um ein Semikolon ";", damit Kommentare angefügt werden können. Das bedeutet, daß Kommentare grundsätzlich mit einem Semikolon beginnen müssen. Die einzelnen Strompfade bzw. Anweisungsfolgen werden außerdem im Kommentar nummeriert. Damit soll das Auffinden von Anweisungen erleichtert werden.

Der Aufruf der Dokumentations-Funktionen kann vom SPS-Interpreter wie auch vom Editor aus erfolgen. Hier wird zunächst der Aufruf über den SPS-Interpreter beschrieben . Die Dokumentations-Funktionen werden im SPS durch Eingabe des Buchstabens "D" für "Dokumentation" gestartet. Danach wird der Bediener über ein Menü geführt. Im folgenden ist ein einfaches Beispiel für den Umgang mit dem Programmtexter abgebildet. Im Gegensatz zum Bildschirm-ausdruck wurden hier Leerzeilen oder Auflistungen von Kommandonamen entfernt, wodurch die Verdeutlichung der Arbeitsschritte nicht beeinflußt wird.

Funktionsbeschreibung

3. Schritt

Nach der Übertragung des SPS-Programms in den Textpuffer des Editors wird dieser durch die Eingabe von "E" über einen Warmstart gestartet. Danach kann die weitere Textbearbeitung vorgenommen werden.

DOKUMENTATION SPS-PROGRAMM V1.8

TEXTPUFFER ==> SPS = 1

SPS ==> TEXTPUFFER = 2

START EDITOR = E

START SPS = S

>E <---- Aufruf des Editors

*** BFZ-EDITOR V1.6 ***

(C) 1989 BFZ, ESSEN

MAT85 = ESC

HILFSMENUE = X

KALTSTART = K

WARMSTART = <CR>, <SPACE>

> <SPACE> <---- Warmstart des Editors durch
Betätigen der SPACE-Taste

EDIT-BUFFER:

START-ADR =8000

STOP -ADR =8077

==> <CR>, <SPACE>

U E 00 ;< <---- in den Textpuffer übertragenes
SPS-Programm

U E 01 ;<

= A 00 ;1<

<

U E 02 ;<

O E 03 ;<

= A 01 ;2<

<

END <---- Ende-Kennung

Die letzten Zeilen im Protokoll zeigen das formatierte SPS-Programm mit Leerzeichen und der Einleitung von Kommentaren durch ein Semikolon nach jeder Anweisung sowie der Nummerierung der zusammengehörenden Anweisungen bzw. Strompfade. Darüber hinaus wird am Programmende ein "END" angehängt, welches im Zusammenhang mit dem Programmlader noch weiter erklärt wird.

4. Schritt

Das Ergebnis einer Kommentierung dieses Programmbeispiels mit dem Editor ist nachfolgend abgebildet. Ist diese Arbeit abgeschlossen, so kann das kommentierte SPS-Programm für den späteren Wiedergebrauch auf Diskette gesichert werden.

Funktionsbeschreibung

```

;*****< <---- überarbeitetes SPS-
;***  A U F G A B E      ***<      Programm im Textpuffer
;*****<
<
;EIN PUMPENMOTOR FUER EINEN FLUESSIGKEITS-<
;BEHAELTER SOLL UEBER ZWEI SCHALTER EIN-<
;GESCHALTET WERDEN<
;DARUEBER HINAUS SIND ZWEI NIVEAU-SCHALTER<
;FUER EIN UNTERES UND EIN OBERES NIVEAU ZU<
;UEBERWACHEN UND BEI BETAETIGUNG IST EINE<
;SIGNALLAMPE EINZUSCHALTEN<
<
;*****<
;***  BELEGUNGS-LISTE    ***<
;*****<
<
; E 00 - SCHALTER 1 PUMPE EIN - SCHLIESSER<
; E 01 - SCHALTER 2 PUMPE EIN - SCHLIESSER<
; E 02 - NIVEAU-SCHALTER 1 - SCHLIESSER<
; E 03 - NIVEAU-SCHALTER 2 - SCHLIESSER<
<
; A 00 - SCHUETZ PUMPE EIN<
; A 01 - SIGNALLAMPE NIVEAU<
<
;*****<
;***  ANWEISUNGS-LISTE  ***<
;*****<
<
U  E 00      ; WENN SCHALTER 1 BETAETIGT IST<
U  E 01      ; UND SCHALTER 2 BETAETIGT IST<
=  A 00      ;1 DANN PUMPE EINSCHALTEN<
<
U  E 02      ; WENN NIVEAU 1 BETAETIGT IST<
O  E 03      ; ODER NIVEAU 2 BETAETIGT IST<
=  A 01      ;2 DANN SIGNALLAMPE EINSCHALTEN<
<
;*****<
;***  PROGRAMM ENDE      ***<
;*****<
<
END<

```

Wenn die Dokumentationsschritte abgeschlossen sind und das kommentierte SPS-Programm auf Diskette gesichert wurde, kann der Editor verlassen und der SPS-Interpreter wieder aufgerufen werden.

5. Schritt

Der Aufruf des SPS-Interpreters aus dem Editor-Betrieb erfolgt durch gleichzeitiges Betätigen der Tasten CONTROL und "Y" (nachfolgend durch CNTL-Y dargestellt). Danach erscheint das bekannte Menü für die Dokumentations-Funktionen. Mit der Eingabe von "S" wird zum SPS-Interpreter verzweigt.

Funktionsbeschreibung

Während der Übertragung werden die SPS-Anweisungen auf korrekte Schreibweise überprüft. Die Übertragung wird abgebrochen, falls ein Fehler auftritt. Fehlerhafte Programmzeilen werden mit der Zeilennummer aufgelistet und die Fehlerstellen werden im Textpuffer des Editors mit einem "?" zum leichteren Auffinden gekennzeichnet. Der SPS-Arbeitsspeicher wird beim Auftreten eines Fehlers sofort wieder gelöscht. War das SPS-Programm korrekt, so meldet der Programmlader "OK" und das Dokumentations-Menü erscheint wieder nach Eingabe von SPACE. Danach kann aus dem Editor durch Eingabe von "S" der SPS-Interpreter gestartet werden.

DOKUMENTATION SPS-PROGRAMM V1.8

```
TEXTPUFFER ==> SPS = 1
SPS ==> TEXTPUFFER = 2
```

```
START EDITOR = E
START SPS    = S
```

```
>S                                <---- Aufruf des SPS-Interpreters
                                   aus dem Editor-Betrieb
```

```
BFZ-SPS-PROGRAMM V2.3 RESTART
```

```
  EDIT
```

```
  GO
```

```
  ...
```

```
  ...
```

```
SPS>
```

Die Überprüfung des Programms im SPS-Arbeitsspeicher erfolgt mit dem LIST-Kommando. Sofern es sich um das zuvor erstellte Programm handelt, erscheinen die nachfolgenden SPS-Anweisungen.

```
SPS> LIST                          <---- Überprüfung des übertragenen
                                   SPS-Programms
```

```
UE00
```

```
UE01
```

```
=A00
```

```
UE02
```

```
OE03
```

```
=A01
```

```
SPS>
```

2.4.2.3 Fehlerbehandlung

Der Programmlader bricht die Übertragung eines SPS-Programms vom Textpuffer des Editors zum SPS-Arbeitsspeicher ab, wenn ein Fehler im SPS-Programm vorliegt. Eine fehlerhafte Programmzeile wird mit der Zeilennummer aufgelistet. Im Textpuffer wird die Fehlerstelle mit einem Fragezeichen "?" überschrieben und kann so leicht gefunden werden. Darüber hinaus wird der SPS-Arbeitsspeicher sofort wieder gelöscht, so daß keine fehlerhaften SPS-Programme gestartet werden können.

Funktionsbeschreibung

In der Testphase eines mit dem Editor erstellten SPS-Programms kann auch eine abschnittsweise Übertragung des SPS-Programms erfolgen, weil die Übertragung stets nur bis zur END-Anweisung im Textpuffer erfolgt. Diese END-Anweisung kann somit Schritt für Schritt im Programm verschoben werden, um den zu übertragenden Programmbereich zu erweitern.

Da mit dem Texteditor auch Assembler-Programme erstellt werden, wird es vorkommen, daß sich ein Assembler-Programm im Textpuffer befindet. Um in diesem Fall eine Übertragung mit dem SPS-Programm- lader zu verhindern, untersucht dieser zunächst den Inhalt des Textpuffers. Falls hier ein Text vorliegt, der am Zeilenanfang nicht mindestens einmal die Operationssymbole "U" und "=" für eine einfache SPS-Anweisungsfolge enthält, so erfolgt keine Übertragung.

Als Trennzeichen im Text sind neben den Leerzeichen auch Tabulator-Steuerzeichen (09H) erlaubt.

Die Operationssymbole UN, ON, =N, =NL dürfen grundsätzlich nicht getrennt werden.

Da der Texteditor den Punkt "." als Steuerzeichen für die Druckerausgabe verwendet, darf auch dieses Zeichen am Anfang einer Zeile stehen. Der Punkt bewirkt beim Ausdruck eines Textes aus dem Textpuffer, daß der Drucker einen Seitenvorschub ausführt. Darüber hinaus lassen sich mit den Punkt-Zeilen sogenannte Kopfzeilen einschließlich einer Seitenummerierung definieren, die auf jeder neuen Seite ausgedruckt werden.

Zusammenfassend können noch folgende Fehlermeldungen auftreten:

Fehlermeldung	Fehlerursache
PROGRAMM ?	kein SPS-Programm im Textpuffer
END ?	SPS-Programm im Textpuffer nicht mit END abgeschlossen
SPS-RAM-ÜBERLAUF	SPS-Programm für den SPS-Arbeitspeicher zu groß
TEXT-RAM ?	kein RAM unter der angegebenen Textpuffer-Adresse
SPS-RAM ?	kein RAM im SPS-Arbeitsspeicherbereich
TEXT-RAM ? GLEICH	Der Text im Textpuffer überschreibt den Inhalt des SPS-Arbeitsspeichers (Anf.-Adr. E000H)

Funktionsbeschreibung

2.5 BFZ-Editor

Der Editor, der speziell für die Assembler-Programmierung vorgesehen ist und durch die Dokumentations-Funktionen auch für die SPS-Programmierung verwendet werden kann, ist dahingehend geändert, daß er in einem neuen Speicherbereich liegt. Damit wird der gleichzeitige Betrieb mit der SPS-Kontaktplanausgabe möglich. Als Anfangsadresse des Textpuffers schlägt der Editor die RAM-Startadresse 8000H vor. Im EPROM befindet sich auch der Text des Hilfsmenüs, der durch Eingabe von CNTL-X auf dem Bildschirm ausgegeben wird.

3. Inbetriebnahme-Hinweise MAT32

Für die Inbetriebnahme der Betriebssoftware im 32-KByte-EPROM vom Typ 27256, bestücken Sie eine 64-KByte-RAM-Baugruppe entsprechend dem Bestückungsplan im Anhang A8. Beachten Sie, daß die erforderlichen Lötbrücken hergestellt werden.

Entgegen der Betriebsweise der 64-KByte-RAM-Baugruppe im CP/M-System, werden die Lötbrücken so hergestellt, daß das 32-KByte-EPROM stets aktiv bleibt, d.h. nicht ausgeblendet wird. Damit soll ein Überschreiben des Speicherbereichs, in dem sich die Betriebssoftware befindet, bei einer Fehlbedienung des Systems verhindert werden. Daher braucht die 64-KByte-RAM-Baugruppe nur mit vier 8-KByte-RAM-Bausteinen bestückt werden.

Da mit der 64-KByte-Speicherbaugruppe der maximale Systemspeicher zur Verfügung steht, darf sich keine weitere Speicherbaugruppe im Baugruppenträger befinden.

Nach dem Einschalten der Betriebsspannung muß sich die Betriebssoftware mit dem Auflisten der MAT85-Kommandos melden. Sollten hier Schwierigkeiten auftreten, so überprüfen Sie die Funktion der Speicherbaugruppe mit Hilfe des BUS-Signalgebers und der BUS-Signalanzeige unter Zuhilfenahme der entsprechenden Fachpraktischen Übungen.

Die Funktion der Baugruppe läßt sich schnell überprüfen, indem die beiden Speicherbereiche von 0000H bis 7FFFH bzw. von 8000H bis FFFFH getestet werden. Im ersten Bereich befindet sich das 32-KByte-EPROM. Dieser Bereich kann lediglich gelesen werden. Ein Überschreiben darf nicht möglich sein. Der zweite Speicherbereich ist der RAM-Bereich. Hier müssen sich die Speicherinhalte verändern lassen. Überprüfen Sie insbesondere nach einem Schreib-/Leseversuch im RAM oberhalb der Adresse C000H erneut den EPROM-Bereich, ob dieser noch aktiv ist. Sollte sich in diesem Bereich der Speicherinhalt überschreiben lassen, so ist die EPROM-Ab-schaltung z.B. durch falsch hergestellte Lötbrücken nicht deaktiviert worden.

Anhang A1: MAT85- und MAT85+ - Kommandos

Nach dem ersten Einschalten des Mikrocomputers ist das Betriebsprogramm MAT85 aktiv. Die erweiterten Funktionen unter MAT85+ können durch Eingabe der SPACE-Taste aufgerufen und auch wieder verlassen werden. Der Kommando-Aufruf erfolgt durch Eingabe des ersten Buchstabens des Kommandonamens gefolgt von der RETURN-Taste (CR). Die Auflistung aller verfügbaren Funktionen erhält man durch Aufruf des HELP-Kommandos.

KMD> HELP

ASSEMBLER
BREAKPOINT
DISASSEMBLER
GO
HELP
IN
LOAD TAPE
MEMORY
NEXT INSTRUCTION
OUT
PRINT
REGISTER
SAVE
TRACE INTERVAL

KMD> <SPACE-Taste>

<---- Umschaltung nach MAT85+
durch Betätigen der SPACE-
Taste

KMD+> HELP

BASIC
COPY
FIND
HELP
INSERT
PROMMER
RAM-TEST
SPS
VERIFY
WRITE CONSTANT

KMD+> <SPACE-Taste>

<---- Rückkehr nach MAT85 durch
erneutes Betätigen der
SPACE-Taste

KMD>

Anhang A2: SPS - Kommandos

Der SPS-Interpreter wird aus dem MAT85+ gestartet. Die SPS-Erweiterungen für die Kontaktplanausgabe und die Dokumentation werden durch Eingabe des Buchstabens "K" bzw. "D" aufgerufen. Diese Funktionen werden in der HELP-Auflistung nicht mit angezeigt.

```

KMD+> SPS                <---- Start des SPS-Interpreters
BFZ-SPS-PROGRAMM V2.3    im MAT85+ oder durch Eingabe
EDIT                     von CNTL-Y im Editor-Betrieb
GO
HELP
LIST
NEW
READ
STEP
TRACE
WRITE
QUIT
<KONTAKTPLAN-AUSGABE>   <---- diese Kommando-Namen werden
<DOKUMENTATION>        <---- nicht angezeigt

```

Die folgenden Aufstellungen geben einen Überblick über die verfügbaren Verknüpfungsanweisungen (Operationen), die möglichen Operanden, sowie die erlaubten Operanden-Adressen.

OPERATION	SYMBOL
UND	U
UND NICHT	UN
ODER	O
ODER NICHT	ON
GLEICH	=
GLEICH NICHT	=N
SETZEN	=S
NICHT SETZEN	=NS
RÜCKSETZEN	=R
NICHT RÜCKSETZEN	=NR
LADEN	=L
NICHT LADEN	=NL

OPERAND	SYMBOL
EINGANG	E
AUSGANG	A
MERKER	M
TIMER (Hardware)	T
ZEITWERK (Software)	Z
COUNTER	C
erlaubte Operanden- adressen	00...07
	10...17
	20...27
	30...37

Anhang A3: Steuer - BASIC - Befehle

Der BASIC-Interpreter wird aus dem MAT85+ gestartet. Im folgenden sind die verfügbaren BASIC-Anweisungen in einer Kurzschreibweise zusammengefaßt.

```
KMD+> BASIC          <---- Start des BASIC-Interpreters
BFZ-STEUER-BASIC V2.4      im MAT85+
```

```
READY
>
```

EDITIER-ANWEISUNGEN:

```
-----
Zeichen löschen: DEL oder BS oder -
Zeile eingeben: <zeilen-nummer> <anweisung> : <anweisung> :...CR
Zeile löschen: <zeilen-nummer> CR
Zeile einfügen: <zwischen-zeilen-nummer>..<anweisungen>...CR
Drucker ein/aus: Ctrl-P
```

SYSTEM-KOMMANDOS:

```
-----
Programm starten:          RUN
Programm listen:          LIST
                          LIST <zeilen-nummer>
Programm löschen:        NEW
Programm von Cassette laden: LOAD
Programm auf Cassette sichern: SAVE
Disketten-Funktionen aufrufen: FLOPPY
Rückkehr zum MAT85+:      QUIT
```

PROGRAMMTEST-KOMMANDOS/ANWEISUNGEN:

```
-----
Einzelschritt ein:       STON
Einzelschritt aus:       STOFF
Tracelauf ein:           TRON
Tracelauf aus:           TROFF
```

KONSTANTE:

```
-----
dezimal:                  -32768.....32767
hexadezimal:              DEC(0000).....DEC(FFFF)
```

```
VARIABLE:                A,B,C,D,E,.....,X,Y,Z
-----
```

```
FELD-VARIABLE:          @(0), @(1), @(2), ..., @(n)
-----
```

```
WERTZUWEISUNGEN:       LET <variable> = <ausdruck>
-----                  <variable> = <ausdruck>
                          LET @( <ausdruck> ) = <ausdruck>
                          @( <ausdruck> ) = <ausdruck>
```

Anhang A3: BASIC - Kommandos

ARITHMETISCHE OPERATIONEN:

```
-----
Addition:          +
Subtraktion:       -
Multiplikation:    *
Division:          /
Klammer auf       (
Klammer zu        )
```

VERGLEICHOPERATIONEN:

```
-----
gleich:            =
größer:           >
kleiner:          <
ungleich:         <>
kleiner gleich:   <=
größer gleich:   >=
logisch UND:      AND
logisch ODER:     OR
logisch NICHT:   NOT
```

BILDSCHIRM-AUSGABE:

```
-----
Bildschirm löschen: CLS
```

```
Textausgabe:      PRINT "<text>"
Konstanten:       PRINT <konstante>;<konstante>...
Variablen:        PRINT <variable>;<variable>.....
Ausdrücke:        PRINT <ausdruck>;<ausdruck>.....
```

Wirkung der Steuerzeichen:

```
Kein Steuerzeichen: PRINT ... in die nächste Zeile gehen
Semikolon:          PRINT ...; in der Position stehenbleiben
Komma:              PRINT ..., in die nächste Zone gehen
```

TASTATUR-EINGABE:

```
-----
Dezimalzahl:      INPUT <variable>
Hexadezimal-Zahl: INPUT #<variable>
Variablen-Liste:  INPUT <variable>,<variable>,...
Melde-Text:       INPUT "<melde-text>",<variable>,...
Einzelzeichen-
Eingabe:          <variable> = $
```

DRUCKER-STEUERUNG:

```
-----
Drucker ein:      LPON
Drucker aus:      LPOFF
```

Anhang A3: BASIC - Kommandos

PORT-EIN-/AUSGABE:

Port-Eingabe: <variable> = INP(<port-nummer>)
Port-Ausgabe: OUT <port-nummer>, <ausgabe-wert>
Warteschleife: WAIT <port-nummer>, <maske>, <vergleichswert>

SPEICHER-ZUGRIFF:

Speicher lesen: <variable> = PEEK(<adresse>)
Speicher schreiben: POKE <adresse>, <speicher-wert>

PROGRAMMVERZWEIGUNGEN:

absolut: GOTO <zeilen-nummer>
bedingt: IF <bedingung> THEN GOTO <zeilen-nummer>
 GOSUB <zeilen-nummer>

UNTERPROGRAMME:

Aufruf: GOSUB <zeilen-nummer>
Rückkehr: RETURN

PROGRAMMSCHLEIFEN:

```
FOR <variable>=<start-wert> TO <stop-wert> STEP <inkrement>
...
<anweisung>
<anweisung>
...
NEXT <variable>
```

FUNKTIONEN:

Betrag: <variable> = ABS(<ausdruck>)
Zufallszahl: <variable> = RND(<grenz-wert>)
Hexadezimal-/
Dezimal-Wandlung: <variable> = DEC(<hexadezimal-zahl>)
Freier Speicher-
bereich: <variable> = FREE

DATEN-/KONSTANTEN-MENGEN

Datenzeile: DATA <konstante>, <konstante>, ...
Daten lesen: READ <variable>, <variable>, ...
Datenzeiger
positionieren: RESTORE <zeilen-nummer>
 RESTORE <variable mit zeilen-nummer>

Anhang A3: BASIC - Kommandos

UNTERPROGRAMM IN MASCHINENSPRACHE:

<variable> = USR(<adresse>, <übergabe-wert>) <ausdruck>

!!
liefert <rückgabe-wert>

vor Aufruf: <übergabe-wert> im HL-Registerpaar
nach Rückkehr: <rückgabe-wert> verknüpft mit <ausdruck>
in <variable>

SONSTIGE ANWEISUNGEN:

Programm-Ende: END
Programm-Stop: STOP
Kommentar: REM <kommentar-text>

Anhang A4: Mini-DOS-Kommandos

Die Disketten-Funktionen des MiniDOS können aus dem MAT85, dem Steuer-BASIC, dem SPS-Interpreter sowie aus dem Editor heraus aufgerufen werden. Davon abhängig werden den Datei-Namen des Benutzers Erweiterungen angehängt, die im Directory anzeigen, über welche Programmoption die Sicherung der Dateien erfolgte. Das MiniDOS-Menü wird mit der Eingabe von Q(UIT) verlassen.

Start des MiniDOS aus Programmteil	Eingabe
MAT85	KMD> F
Steuer-BASIC	>FLOPPY
SPS	SPS> F
Editor	CNTL-S

KMD > F

<---- Start des Mini-DOS im MAT85

BFZ-MINI-DOS V1.4

(C) 1985 BY BFZ, ESSEN, W. GERMANY

MENUE:

DIRECTORY
 ERASE
 FORMAT
 LOAD
 SAVE
 QUIT

BITTE BUCHSTABE EINGEBEN:

Anhang A5: EDITOR - Kommandos

Der Editor wird mit der Eingabe von "E" im MAT85-Betrieb gestartet. Darüber hinaus kann er über die Dokumentations-Funktionen im SPS-Betrieb aufgerufen werden.

```
KMD> E                <---- Start des Editors im MAT85
*** BFZ-EDITOR V1.6 ***
(C) 1989 BFZ, ESSEN
```

```
MAT85 = ESC
HILFSMENUE = X
KALTSTART = K
WARMSTART = <CR>,<SPACE>
```

Die für die Textverarbeitung zur Verfügung stehenden Editier-Kommandos sind im folgenden Hilf-Menü zusammengefaßt. Dieses Hilfs-Menü kann während der Textbearbeitung stets wieder aufgerufen werden, wenn eine bestimmte Funktion und deren Aufruf gesucht wird. In der Darstellung bedeutet die Zeichenkombination "^" gefolgt von einem Buchstaben, daß gleichzeitig mit der Buchstaben-Taste die CONTROL-Taste zu betätigen ist.

```
*****
*** HILFS-MENUE ***
*****
```

----- VERARBEITUNG -----

```
^X = HILFS-MENUE          ^S = SPEICHERN/LADEN
^N = NEUSTART             ^D = DRUCKER-MENUE
                          ^A = ASSEMBLIEREN
                          ^Y = SPS-PROGRAMM-LADER/TEXTER
```

---- ZEICHEN EDITIEREN --- --- ZEILEN EDITIEREN ---

```
^R = ZEICHEN RADIEREN      ^ZL = ZEILE LOESCHEN
DEL = ZEICHEN LINKS LOESCHEN ^ZE = ZEILE EINFUEGEN
^E = EINFUEGEN EIN/AUS    ^ZA = ZEILE ANHAENGEN
```

---- CURSOR-STEUERUNG --- --- SEITEN BLAETTERN ---

```
PFEILE = CURSOR-STEUERUNG    ^U = SEITE UNTEN
^CO = AN OBEREN BILDRAND     ^O = SEITE OBEN
^CU = AN UNTEREN BILDRAND    ^CA = CURSOR ZUM TEXTANFANG
^CL = AN LINKEN BILDRAND     ^CE = CURSOR ZUM TEXTENDE
^CR = AN RECHTEN ZEILENRAND
^W = AUF NAECHSTES WORT      ^T = TABULATOR
```

--- BLOCK-BEFEHLE --- --- FINDE / ERSETZE ---

```
^BM = BLOCK MARKIEREN       ^FS = SETZE SUCHWORT
^BR = B.-MARK. RUECKSETZEN  ^FF = FINDE SUCHWORT
^BF = B.-MARK. FINDEN       ^FE = ERSETZE SUCHWORT
```

```
^BL = BLOCK LOESCHEN
^BK = BLOCK KOPIEREN
^BV = BLOCK VERSCHIEBEN
^BS = BLOCK SPEICHERN (FLOPPY/CASSETTE)
```

Anhang A6: SPS - Programmbeispiel "Anweisungsliste"

```

;-----;
; B F Z   E S S E N   ! AUTOMATISIEREN MIT MFA-SPS ;
;                                     ! MIKROCOMPUTER ;
;-----;
; BEISPIEL:           ! DATUM : 14.04.1989 ;
;                                     ! AEND. : ;
;   WENDESCHUETZ-    ! BEARB.: N.N. ;
;   STEUERUNG        ! ;
;-----;

```

```

;*****
;***   A U F G A B E   ***
;*****

```

```

;EIN DREHSTROMMOTOR, DER IN EINEM HYDRAULIK-
;SYSTEM EINGEBUNDEN IST, WIRD UEBER DREI
;HANDTASTER IM TIPP-BETRIEB IN DER DREH-
;RICHTUNG GESTEUERT.
;DREHRICHTUNGSUMKEHR IST NUR AUS DEM STILL-
;STAND ERLAUBT.
;DER MOTOR DARF NICHT EINGESCHALTET WERDEN,
;WENN DIE TEMPERATUR DES KUEHLSYSTEMS BZW.
;WENN DER GRENZDRUCK DES HYDRAULIK-SYSTEMS
;ERREICHT IST.

```

```

;*****
;***   BELEGUNGS-LISTE   ***
;*****

```

```

; E 00 - HANDTASTER RECHTSLAUF - SCHLIESSER
; E 01 - HANDTASTER LINKSLAUF - SCHLIESSER
; E 02 - HANDTASTER AUS - OEFFNER
; E 03 - TEMPERATUR-FUEHLER - OEFFNER
; E 04 - DRUCK-SCHALTER - OEFFNER

; A 00 - SCHUETZ RECHTSLAUF
; A 01 - SCHUETZ LINKSLAUF

```

Anhang A6: SPS - Programmbeispiel "Anweisungsliste"

```
*****
***   PROGRAMM - ANFANG   ***
*****

;*** AUSSCHALTBEDINGUNGEN ***

UN  E 02   ; WENN HANDTASTER AUS BETAETIGT
ON  E 03   ; ODER TEMPERATUR ZU GROSS
ON  E 04   ; ODER GRENZDRUCK ERREICHT
=   M 00   ; DANN EINSCHALTEN VERHINDERN

;***   RECHTSLAUF EIN   ***

U   E 00   ; WENN RECHTSLAUF BETAETIGT
UN  M 00   ; UND KEINE AUSSCHALT-BEDINGUNG
UN  A 01   ; UND NICHT IM LINKSLAUF
=S  A 00   ; DANN MOTOR EIN

;***   RECHTSLAUF AUS   ***

U   M 00   ; WENN AUSSCHALT-BEDINGUNG
=R  A 00   ; DANN MOTOR AUS

;***   LINKSLAUF EIN   ***

U   E 01   ; WENN LINKSLAUF BETAETIGT
UN  M 00   ; UND KEINE AUSCHALT-BEDINGUNG
UN  A 00   ; UND NICHT IM RECHTSLAUF
=S  A 01   ; DANN MOTOR EIN

;***   LINKSLAUF AUS   ***

U   M 00   ; WENN AUSSCHALT-BEDINGUNG
=R  A 01   ; DANN MOTOR AUS

*****
***   PROGRAMM ENDE   ***
*****
```

Anhang A6: SPS - Programmbeispiel "Kontaktplan"

```

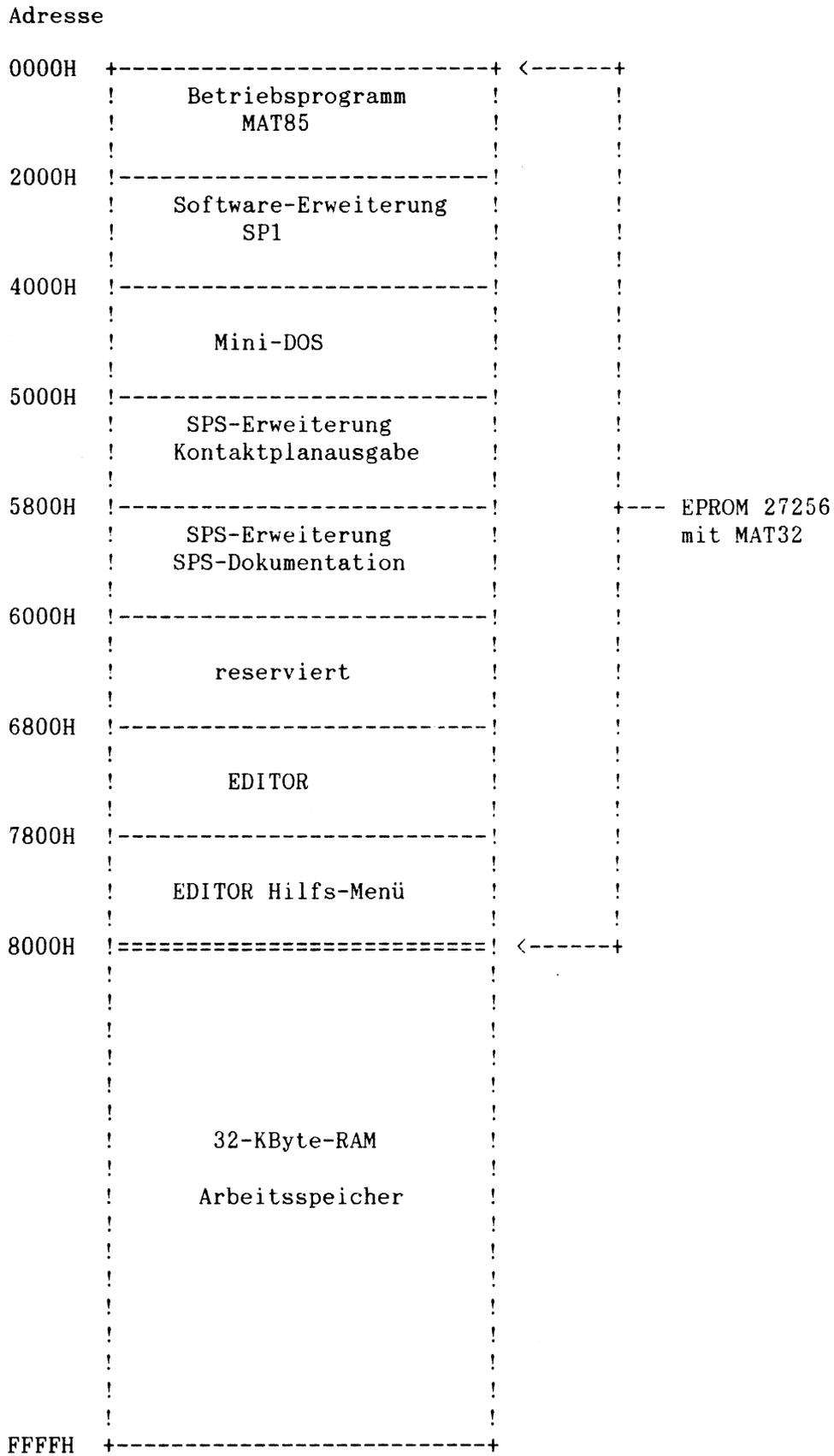
I
I----I/I-----+
I  E02          I  I
I              I  I
I----I/I-----+
I  E03          I  I
I              I  I
I----I/I-----+----( )----I
I  E04          M00  I
I              I
I----I I--I/I--I/I-----+----(S)----I
I  E00 M00 A01  A00  I
I              I
I----I I-----+----(R)----I
I  M00          A00  I
I              I
I----I I--I/I--I/I-----+----(S)----I
I  E01 M00 A00  A01  I
I              I
I----I I-----+----(R)----I
I  M00          A01  I
I              I

```

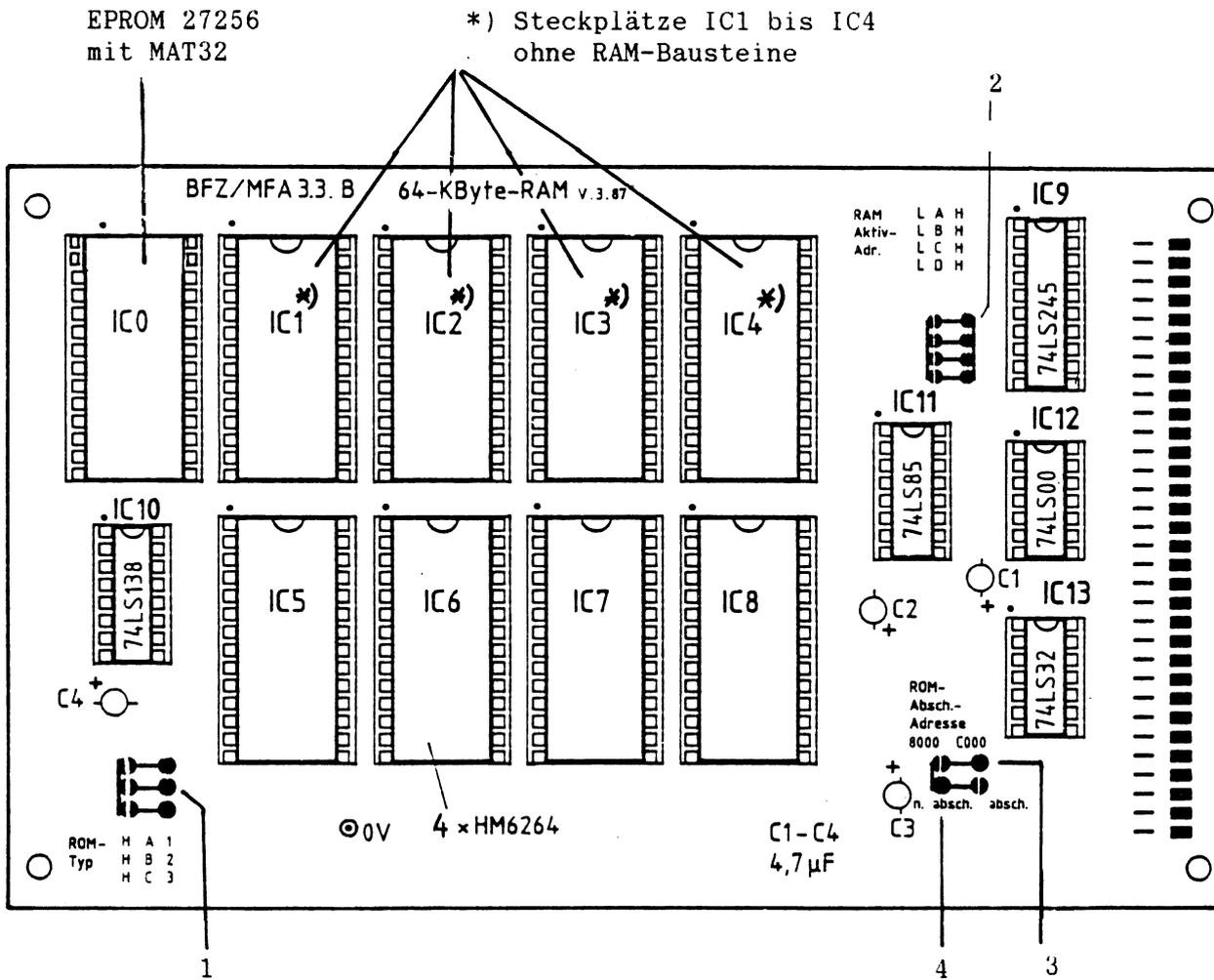
Bemerkung:

Anstelle der eckigen Klammern für die Kontakte wird in dieser Darstellung das Zeichen "I" verwendet. Die eckigen Klammern werden sonst auf einem Drucker, der auf den deutschen ASCII-Zeichensatz eingestellt ist, als Umlaute gedruckt.

Anhang A7: Speicherbelegung



Anhang A8: Bestückungsplan 64-KByte-RAM



Hinweis:

Für den Betrieb des 32-KByte-EPROMs sind die in der nachfolgenden Tabelle aufgeführten Lötbrücken erforderlich.

Pos.	Funktion	geschlossene Brücken
1	ROM-Typ 27256	A - 1 B - 2 C - 3
2	RAM-Aktiv-Bereich ab 8000H	A - H B - H C - H D - H
3	ROM-Abschalt-Adresse (unbedeutend)	C000 oder (8000)
4	ROM nicht abschaltbar oder abschaltbar	n. absch.

Inhaltsverzeichnis

	Seite
1. Einleitung	1
1.1. Aufgaben und Funktionsmerkmale des BFZ-EDITORS	1
1.2. Installation und Inbetriebnahme	3
2. Überblick über die Steuerbefehle des EDITORS	7
3. Aufruf des BFZ-EDITORS	8
4. Beschreibung der Steuerbefehle des BFZ-EDITORS	11
4.1. Cursor-Steuerung mit Hilfe der "Pfeil"-Tasten und <CR>	11
4.2. Neustart: CTRL-N	12
4.3. Aufruf des Hilfsmenues: CTRL-X (Übung 1, Übung 2)	12
4.4. Zeichen radieren, Zeichen links löschen	16
4.5. Einfüge-Modus Ein-/Aus	17
4.6. Zeile löschen, Zeile einfügen, Zeile anhängen , (Übung 3)	17
4.7. Erweiterte Cursor-Bewegungen	20
4.8. Tabulator	20
4.9. Seiten blättern	23
4.10. Finden eines "Text-Strings", Ersetzen eines "Text-Strings"	24
4.11. Block-Befehle	26
4.11.1. Block markieren, Block-Markierungen rücksetzen, Block-Markierungen finden	26
4.11.2. Block löschen, Block kopieren, Block verschieben	27
4.12. Speichern und Laden mit Hilfe von Diskette oder Cassette	31
4.12.1. Speichern auf Diskette oder Cassette	31
4.12.2. Laden einer Textdatei von Diskette oder Cassette	34
4.13. Drucken des EDIT-Buffer-Inhalts	36
4.13.1. Drucker-Menue	36
4.13.2. Steuerung des Formularvorschubs mit ".", Ausdruck von Kopfzeilen	38
4.14. Assemblieren	40
Anhang A: Lösung der Übung 1	
Anhang B: Anpassung des BFZ-EDITORS an andere Terminals	
Anhang C: Aufbau des Hilfs-Menues	

1. Einleitung

1.1. Aufgaben und Funktionsmerkmale des BFZ-EDITORS

Die Aufgabe eines EDITOR-Programms (EDITOR = Redakteur, Schriftleiter) besteht darin, mit Hilfe eines Computers Texte zu erstellen, zu bearbeiten und zu verarbeiten. Bei diesen Texten kann es sich z.B. um Briefe, Beschreibungen, Listen oder ähnliches handeln. Von besonderer Bedeutung für die Computertechnik sind Programm-Texte in Assemblersprache (der sogenannte "Source- oder Quellcode"). Lassen sich diese Programm-Texte mit Hilfe eines EDITORS bearbeiten, auf Diskette oder Cassette speichern und anschließend assemblieren, so sind spätere Änderungen oder Ergänzungen eines Programms leicht und komfortabel möglich. Aus diesem Grund eignet sich der BFZ-EDITOR besonders dazu, Assembler-Quell-Programme komfortabel zu erstellen, zu bearbeiten und anschließend durch den Assemblier-Vorgang in Maschinensprache umzusetzen (8085-Assembler-Code).

Die Arbeitsweise eines EDITOR-Programms besteht darin, die vom Bediener eingegebenen Textzeichen auf dem Bildschirm anzuzeigen und zusätzlich im Arbeitsspeicher des Computers zu speichern und zu verwalten. Der Speicherbereich, in dem die Textzeichen abgelegt werden, ist der sogenannte "EDIT-Buffer" (Buffer = Puffer, Zwischenspeicher). Die Bearbeitung des Textes mit Hilfe des BFZ-EDITORS erfolgt "Bildschirm-orientiert", d.h., auf dem Bildschirm wird das genaue Abbild des EDIT-Buffer-Inhalts zeilenweise dargestellt. Jede Veränderung von Textzeichen innerhalb des EDIT-Buffers wird direkt auf dem Bildschirm angezeigt. Dazu gehören z.B.:

- Positionierung des "Cursors" auf ein bestimmtes Textzeichen
- Positionierung des "Cursors" an den oberen, unteren, linken oder rechten Textrand, sowie auf das nächste "Wort"
- Löschen, Einfügen, Trennen oder Anhängen von Textzeilen
- "Blättern" auf die nächste oder vorherige Textseite, sowie an den Textanfang oder an das Textende
- Markieren, Kopieren, Verschieben oder Löschen von "Textblöcken"
- Suchen und Ersetzen bestimmter Zeichenketten ("Strings")

Weitere Funktionsmerkmale des BFZ-EDITORS sind:

- Bediener-freundliches Speichern und Laden des EDIT-Buffer-Inhalts mit Hilfe von Diskette oder Cassette
- Speichern und Laden von markierten Textblöcken
- Anbinden von gespeicherten Texten an den aktuellen EDIT-Buffer-Inhalt

- Drucken des bearbeiteten Textes mit Voreinstellung des linken Zeilenrandes, der maximalen Zeilenzahl pro Druckseite, sowie Steuerung des Formular-Vorschubs und wahlweisem Ausdruck einer Kopfzeile mit automatischer Seitennumerierung
- Assemblieren von 8085-Assembler-Programmen, wahlweise mit oder ohne Bildschirm-Ausdruck, bzw. Drucker-Listing
- Möglichkeit der Weiterverarbeitung des erstellten Textes mit Hilfe eines anderen Textverarbeitungs-Programms (z.B. "WORD-STAR")

Hinweis:

Das MFA-Video-Interface stellt alle verwendeten Buchstaben in Großschreibung dar. Im EDIT-Buffer werden aber alle Buchstaben so abgelegt, wie sie auch tatsächlich mit der Tastatur eingegeben wurden (z.B. Tastatur-Eingabe "a" ==> "a" wird als ASCII-Zeichen (= 61 hex) im EDIT-Buffer abgelegt ==> auf dem Bildschirm erscheint "A"). Beim Ausdrucken des EDIT-Buffer-Inhaltes werden aber alle als Kleinbuchstaben eingegebenen Zeichen auch als Kleinbuchstaben gedruckt, so daß der gedruckte Text u.U. bzgl. der Groß- und Kleinschreibung nicht mit dem auf dem Bildschirm dargestellten Text übereinstimmt. Soll auch die Ausgabe auf dem Drucker nur in Großschreibung erfolgen, so muß bei der Texteingabe die Taste

"ALPHA-LOCK"

aktiviert werden (Ausnahme: der Ausdruck von Programm-Listings beim Assemblieren erfolgt immer in Großschreibung, siehe 4.14).

1.2. Installation und Inbetriebnahme

Der BFZ-EDITOR ist ein Maschinenprogramm mit einer Länge von 4 KByte. Es ist in zwei 2-KByte-EPROMs vom Typ 2716 gespeichert und belegt den Adreßraum von Adresse 5000 bis 5FFF (hex). Das sogenannte "Hilfs-Menue" mit einer Länge von 1440 Byte (entspr. 05A0 hex) ist in einem weiteren 2-KByte-EPROM 2716 gespeichert. Es besteht im wesentlichen aus einem Text, der die Steuerzeichen des BFZ-EDITORS beschreibt, und kann wahlweise mitbenutzt werden, um das Arbeiten mit dem BFZ-EDITOR gerade in den Anfangsphasen zu erleichtern.

Je nach vorliegendem Software-Ausbau des vorhandenen Systems ergeben sich unterschiedliche Installationsmöglichkeiten für das EDITOR-Programm in Verbindung mit dem "Hilfs-Menue":

a) Vorhandener Software-Ausbau: MAT-85

Adresse (hex)	Software
0000 - 07FF	MAT-85
0800 - 0FFF	MAT-85
1000 - 17FF	MAT-85
1800 - 1FFF	MAT-85
2000 - 27FF	
2800 - 2FFF	
3000 - 37FF	
3800 - 3FFF	
4000 - 47FF	EDITOR-Hilfs-Menue (wahlweise)
4800 - 4FFF	
5000 - 57FF	BFZ-EDITOR (1)
5800 - 5FFF	BFZ-EDITOR (2)

Aufruf des BFZ-EDITORS von der Kommando-Ebene aus:

KMD> G(o)	G	<CR>	oder	<SPACE>	eintippen
START-ADR = 5000	5000	<CR>	oder	<SPACE>	eintippen

Hinweis: <CR> = Carriage-Return-Taste
<SPACE> = Leer-Taste

b) Vorhandener Software-Ausbau: MAT-85 und Software-Paket SP1

Adresse (hex)	Software
-----	-----
0000 - 07FF	MAT-85
0800 - 0FFF	MAT-85
1000 - 17FF	MAT-85
1800 - 1FFF	MAT-85
-----	-----
2000 - 27FF	SP1
2800 - 2FFF	SP1
3000 - 37FF	SP1
3800 - 3FFF	SP1
-----	-----
4000 - 47FF	EDITOR-Hilfs-Menue (wahlweise)
4800 - 4FFF	
5000 - 57FF	BFZ-EDITOR (1)
5800 - 5FFF	BFZ-EDITOR (2)
-----	-----

Aufruf des BFZ-EDITORS von der Kommando-Ebene aus:

mit EDITOR-Hilfs-Menue:

KMD> E

E <CR> oder <SPACE> eintippen

ohne EDITOR-Hilfs-Menue:

KMD> G(o)
START-ADR = 5000

G <CR> oder <SPACE> eintippen
5000 <CR> oder <SPACE> eintippen

c) Vorhandener Software-Ausbau: MAT-85 , Software-Paket SP1 und BFZ-MINI-DOS

Adresse (hex)	Software
0000 - 07FF	MAT-85
0800 - 0FFF	MAT-85
1000 - 17FF	MAT-85
1800 - 1FFF	MAT-85
2000 - 27FF	SP1
2800 - 2FFF	SP1
3000 - 37FF	SP1
3800 - 3FFF	SP1
4000 - 47FF	BFZ-MINI-DOS
4800 - 4FFF	BFZ-MINI-DOS
5000 - 57FF	BFZ-EDITOR (1)
5800 - 5FFF	BFZ-EDITOR (2)

Aufruf des BFZ-EDITORS von der Kommando-Ebene aus:

KMD> E E <CR> oder <SPACE> eintippen

oder

KMD> G(o)
START-ADR = 5000 G <CR> oder <SPACE> eintippen
5000 <CR> oder <SPACE> eintippen

Soll in dieser Konfiguration das EDITOR-Hilfs-Menue mitbenutzt werden, so muß der Inhalt des EPROMs "HELP.MEN" auf eine Diskette gespeichert werden. Beim Arbeiten mit dem BFZ-EDITOR läßt sich das Hilfs-Menue dann an jede beliebige freie Stelle in den Arbeitsspeicher laden und steht dort dem BFZ-EDITOR zur Verfügung.

Anleitung zum Kopieren des EDITOR-Hilfs-Menues auf Diskette:

1. Möglichkeit:

Laden Sie den Inhalt des EPROMs "HELP.MEN" mit Hilfe der EPROM-Programmier-Baugruppe in den Arbeitsspeicher (z.B. E000 - E7FF). Speichern Sie den notwendigen Inhalt des EPROMs mit Hilfe des BFZ-MINI-DCS auf die Diskette:

SAVE

START-ADR = E000
STOP -ADR = E59F

NAME: HELP.MEN

2. Möglichkeit:

Stecken Sie das EPROM "HELP.MEN" in einen freien EPROM-Steckplatz (z.B. 5000 - 57FF). Speichern Sie den notwendigen Inhalt des EPROMs mit Hilfe des BFZ-MINI-DOS auf die Diskette:

SAVE

START-ADR = 5000
STOP -ADR = 559F

NAME: HELP.MEN

2. Überblick über die Steuerbefehle des EDITORS

Die nachfolgende Liste stellt alle Steuerbefehle des BFZ-EDITORS dar. Die Wirkung dieser Steuerbefehle wird dem Bediener im nachfolgenden erläutert.

```

*****
***  HILFS-MENUE  ***
*****

----- VERARBEITUNG -----

^X = HILFS-MENUE           ^S = SPEICHERN/LADEN
^N = NEUSTART              ^D = DRUCKER-MENUE
                           ^A = ASSEMBLIEREN

---- ZEICHEN EDITIEREN ----      --- ZEILEN EDITIEREN ---

^R = ZEICHEN RADIEREN          ^ZL = ZEILE LOESCHEN
DEL = ZEICHEN LINKS LOESCHEN  ^ZE = ZEILE EINFUEGEN
^E = EINFUEGEN EIN/AUS        ^ZA = ZEILE ANHAENGEN

---- CURSOR-STEUERUNG ----      --- SEITEN BLAETTERN ---

PFEILE = CURSOR-STEUERUNG      ^U = SEITE UNTEN
^CO = AN OBEREN BILDRAND        ^O = SEITE OBEN
^CU = AN UNTEREN BILDRAND      ^CA = CURSOR ZUM TEXTANFANG
^CL = AN LINKEN BILDRAND       ^CE = CURSOR ZUM TEXTENDE
^CR = AN RECHTEN ZEILENRAND    -----
^W = AUF NAECHSTES WORT        ^T = TABULATOR

--- BLOCK-BEFEHLE ---          --- FINDE / ERSETZE ---

^BM = BLOCK MARKIEREN          ^FS = SETZE SUCHWORT
^BR = B.-MARK. RUECKSETZEN    ^FF = FINDE SUCHWORT
^BF = B.-MARK. FINDEN        ^FE = ERSETZE SUCHWORT

^BL = BLOCK LOESCHEN
^BK = BLOCK KOPIEREN
^BV = BLOCK VERSCHIEBEN
^BS = BLOCK SPEICHERN (FLOPPY/CASSETTE)

```

Erläuterungen:

^N = CTRL-N , d.h. CTRL-Taste drücken und festhalten, "N" eingeben, CTRL-Taste loslassen

^ZL = CTRL-ZL , d.h. CTRL-Taste drücken und festhalten, "Z" eingeben, "L" eingeben, CTRL-Taste loslassen

Wird bei einer Doppel-CTRL-Sequenz (z.B. ^ZL) als zweites Zeichen ein ungültiger Buchstabe eingegeben, so führt dieses zu einem Abbruch der eingeleiteten CTRL-Funktion mit einer gleichzeitigen akustischen Fehlermeldung. So lassen sich irrtümlich eingeleitete Doppel-CTRL-Sequenzen durch die Eingabe von <SPACE> als zweites Zeichen gezielt abbrechen (z.B. CTRL-Z <SPACE>).

3. Aufruf des BFZ-EDITORS

Je nach Installation des BFZ-EDITORS (siehe 1.2.) wird dieser von der Kommando-Ebene aus aufgerufen:

- | | | | | | | |
|----|--|------|------|------|---------|-----------|
| a) | <code>KMD> E</code> | E | <CR> | oder | <SPACE> | eintippen |
| b) | <code>KMD> G(o)</code>
<code>START-ADR = 5000</code> | G | <CR> | oder | <SPACE> | eintippen |
| | | 5000 | <CR> | oder | <SPACE> | eintippen |

Nach dem EDITOR-Aufruf wird der Bildschirm gelöscht, und es erscheint folgender Meldetext:

```

*** BFZ-EDITOR V1.3 ***

(C) 1987 BFZ, ESSEN

MAT85 = ESC
HILFSMENUE = X
KALTSTART = K
WARMSTART = <CR>, <SPACE>

> _

```

Durch Betätigen der Taste

`ESC`

(Escape = Flucht)

erfolgt eine Rückkehr zum Betriebsprogramm MAT-85. Während der Bearbeitung des Textes (Editieren) ist die "ESC"-Taste inaktiv, so daß die Rückkehr zum Betriebssystem immer mit Hilfe eines "Neustarts" (CTRL-N, siehe 4.2.) mit anschließendem "ESC" erfolgen sollte.

Eine unbeabsichtigte Rückkehr zum Betriebsprogramm MAT-85 durch versehentliche Betätigung der "ESC"-Taste führt nicht zu einem Verlust des bearbeiteten Textes. Durch einen erneuten Aufruf des EDITORS mit nachfolgendem "Warmstart" kann die Textbearbeitung fortgesetzt werden.

Das Hilfsmenue läßt sich aufrufen, indem die Taste

X

(X = Hilfsmenue)

gedrückt wird. Das EDITOR-Programm durchsucht in diesem Fall den Arbeitsspeicher nach den Identifizierungs-Bytes des Hilfsmenues (ED DD ED, hex) und erkennt so, ob das Hilfsmenue installiert wurde.

Wird die Hilfsmenue-Kennung gefunden, so erscheint nach dem Löschen des Bildschirms die erste Seite des Menues. Durch Betätigen der <SPACE>-Taste läßt sich die zweite und dritte Seite des Hilfsmenues auf dem Bildschirm darstellen. Durch nochmalige Betätigung der <SPACE>-Taste beginnt die Darstellung des Hilfsmenues erneut mit der ersten Hilfsmenue-Seite ("Endlos-Blättern"). Mit Hilfe der <CR>-Taste kann die Darstellung des Hilfsmenues unterbrochen werden, und es erscheint wieder der EDITOR-Meldetext auf dem Bildschirm.

Wird das Hilfsmenue nicht gefunden, so bleibt der Bildschirminhalt unverändert, und es ertönt als Bedienerhinweis ein Klingelzeichen ("Bell").

Soll der EDIT-Buffer "gelöscht" werden, um einen neuen Text zu bearbeiten oder um einen auf Diskette oder Cassette gespeicherten Text in den "leeren" EDIT-Buffer zu laden, so muß ein "Kaltstart" ausgeführt werden:

K

(K = Kaltstart)

Wird dagegen ein "Warmstart" ausgeführt, so bleibt der aktuelle Text des EDIT-Buffers erhalten:

<CR> oder <SPACE>

(= Warmstart)

In beiden Fällen (Kaltstart oder Warstart) erscheint nun die Meldung:

**EDIT-BUFFER:
START-ADR =6000 _**

Die Startadresse 6000 wird immer dann vorgeschlagen, wenn das MFA-System neu eingeschaltet wurde ("Power-On-Reset"). Diese Vorschlagadresse kann durch <CR> oder <SPACE> angenommen oder durch Eintippen einer neuen Startadresse mit anschließendem <CR> oder <SPACE> verändert werden. Wichtig ist auf jeden Fall, daß ab der angegebenen Startadresse ein freier Speicherbereich für den EDIT-Buffer vorhanden ist, der mit dem zu bearbeitenden Text gefüllt werden kann.

Das EDITOR-Programm führt nun für die angegebene Start-Adresse einen "RAM"-Test durch. Steht an dieser Stelle kein Schreib-Lese-Speicher zur Verfügung, so wird erneut die Eingabe einer neuen Start-Adresse durch den Bediener verlangt. Zusätzlich erfolgt eine akustische Fehlermeldung.

War der "RAM"-Test erfolgreich, so wird anschließend der EDIT-Buffer ab der angegebenen Start-Adresse nach dem Markierungsbyte für das EDIT-Buffer-Ende (00) durchsucht. Wird dieses gefunden, so wird die Meldung über die Länge des EDIT-Buffers durch die Angabe der Stop-Adresse vervollständigt, z.B. bei einem Kaltstart:

```

EDIT-BUFFER:
  START-ADR =6000
  STOP -ADR =6002

==> <CR>,<SPACE>

```

(Bemerkung: Bei einem Kaltstart werden an den Anfang des EDIT-Buffers die beiden Steuerbytes "0D 0A" (hex), sowie die Buffer-Ende-Markierung "00" eingeschrieben. Als EDIT-Buffer-Ende ergibt sich also für obiges Beispiel die "Stop-Adresse" 6002 (hex), siehe auch Übung 1 im Kapitel 4.)

Bei einem Warmstart wird je nach Länge des bearbeiteten Textes die entsprechend veränderte Stop-Adresse angegeben. Die Länge des EDIT-Buffers ist beliebig, sie wird nur durch den zur Verfügung stehenden freien Bereich im Arbeitsspeicher eingeschränkt. Hierbei muß der Anwender unbedingt darauf achten, daß keine wichtigen Speicherbereiche durch den eingegebenen Text überschrieben werden, z.B.:

```

!!!      ab F800:  Daten-Puffer für BFZ-MINI-DOS      !!!
!!!      ab FB00:  Variablen-Bereich für BFZ-EDITOR   !!!

```

Nach einem Warmstart wird der Bildschirm gelöscht, und die erste Seite des EDIT-Buffer-Inhalts erscheint auf dem Bildschirm. Der im EDIT-Buffer befindliche Text kann weiter bearbeitet werden.

Nach einem Kaltstart ist der Bildschirm leer. Hier kann nun die Eingabe eines neuen Textes erfolgen.

4. Beschreibung der Steuerbefehle des BFZ-EDITORS

4.1. Cursor-Steuerung mit Hilfe der "Pfeil"-Tasten und <CR>

Nach dem Kalt- oder Warmstart des EDITOR-Programms kann der EDIT-Buffer mit Hilfe der Tastatureingabe des Bedieners bearbeitet werden. Der Text wird zeilenweise eingegeben und jede Zeile mit einem <CR> abgeschlossen. Als Kennzeichnung des Zeilenendes erscheint das Zeichen "<" auf dem Bildschirm. Die maximale Zeilenlänge für das MFA-Video-Interface liegt bei 61 Zeichen. Wird diese maximale Zeilenlänge erreicht, so können keine weiteren Textzeichen eingegeben werden. In diesem Fall ertönt als Hinweis für den Bediener ein Klingelzeichen. Um die Textbearbeitung in der nächsten Zeile fortsetzen zu können, muß ein <CR> eingegeben werden.

Die eingegebenen Text- und Satzzeichen werden vom EDITOR im EDIT-Buffer gespeichert und verwaltet. Als Kennzeichnung der Zeilentrennung stehen im EDIT-Buffer an der entsprechenden Stelle die beiden Steuer-Bytes "0D 0A" (hex). Diese Steuer-Bytes besitzen im internationalen ASCII-Code folgende Bedeutung:

0D = Carriage-Return (Wagen-Rücklauf, d.h. Cursor an den Zeilenanfang)

0A = Line-Feed (Zeilenvorschub, d.h. Cursor eine Zeile nach unten)

Wurde eine Zeile mit <CR> abgeschlossen und das Zeilenende mit "<" auf dem Bildschirm markiert, so kann dieses markierte Zeilenende nicht mehr überschrieben werden. Mit Hilfe des "Einfüge-Modus" läßt sich allerdings das markierte Zeilenende bis auf die maximale Zeilenlänge nach rechts verschieben (siehe "Einfüge-Modus" 4.5.).

Mit Hilfe der "Pfeil"-Tasten zur Cursor-Steuerung (auf der Tastatur des Video-Interface links und rechts neben der SPACE-Taste) läßt sich der Cursor über den erstellten Text auf dem Bildschirm bewegen.

Die Cursorbewegung nach links bzw. nach rechts wird jeweils gestoppt, wenn der linke bzw. rechte Textrand erreicht wird (Klingelzeichen als Hinweis für den Bediener).

Die Cursorbewegung nach oben bzw. unten erfolgt innerhalb der aktuellen Bildschirm-Spalte. Ist die neu erreichte Zeile allerdings kürzer als die "alte" Zeile, so wird der Cursor an das Ende der neuen Zeile geführt.

Eine Aufwärtsbewegung des Cursors endet am oberen Bildschirmrand (Klingelzeichen als Bedienerhinweis). Um den davor liegenden Text zu erreichen, muß um eine "Seite nach oben" geblättert werden (siehe auch 4.9.: "Seiten blättern").

Ist der erstellte Text länger als eine Bildschirmseite, so wird der Bildschirminhalt jeweils um eine Zeile nach oben verschoben ("Scrollen"), wenn eine Abwärtsbewegung des Cursors aus der untersten Zeile erfolgt (Cursor-Pfeil "unten"). Anschließend wird der Cursor auf den Anfang der neuen, untersten Bildschirmzeile geführt.

4.2. Neustart: CTRL-N

Alle in 2. angegebenen Steuerzeichen des EDITORS können vom Bediener eingegeben werden, während der EDIT-Buffer bearbeitet wird, der Inhalt des Buffers also auf dem Bildschirm dargestellt wird. Um die Eingabe von Steuerzeichen von der Eingabe von Text- und Satzzeichen zu unterscheiden, müssen alle Steuerzeichen mit Hilfe der CTRL-Taste eingegeben werden (Control). Dazu wird vom Bediener die CTRL-Taste gedrückt und festgehalten, dann das entsprechende Steuerzeichen eingegeben und beide Tasten wieder losgelassen. Die Eingabe mit Hilfe der CTRL-Taste wird vielfach auch wie folgt dargestellt, z.B.:

CTRL-N = ^N	(N = Neustart)
-------------	----------------

Wird während der Textbearbeitung "CTRL-N" eingegeben, so wird der Bildschirm gelöscht, und es erscheint der EDITOR-Meldetext. Der Bediener hat nun die Wahl, einen Kalt- oder einen Warmstart auszuführen und die Startadresse des EDIT-Buffers unter Umständen zu verändern.

Wird ein Warmstart mit derselben Startadresse durchgeführt, erhält der Anwender auf diesem Wege eine Information über die derzeitige Länge des EDIT-Buffers. Dieses ist unter Umständen wichtig, um ein versehentliches Überschreiben anderer wichtiger Speicherinhalte durch den EDIT-Buffer zu vermeiden.

Mit Hilfe eines Kaltstarts mit derselben Startadresse läßt sich der aktuelle Buffer-Inhalt "löschen". Mit einem Kaltstart unter einer anderen Startadresse kann der Bediener einen neuen EDIT-Buffer "eröffnen" und unabhängig vom ersten Buffer bearbeiten. Die Umschaltung zwischen den einzelnen EDIT-Buffern läßt sich dann durch einen Warmstart mit der jeweiligen Buffer-Startadresse durchführen.

4.3. Aufruf des Hilfsmenues: CTRL-X

Auch während der Textbearbeitung läßt sich das Hilfsmenue aufrufen (siehe auch 3.):

CTRL-X	(X = Hilfsmenue)
--------	------------------

Nach der Rückkehr aus dem Hilfsmenue wird der alte Bildschirminhalt wieder hergestellt, und die Textbearbeitung kann in der alten Bildschirmzeile fortgeführt werden.

Übung 1

EDITOR-Aufruf, einfache Texteingabe, Cursor-Steuerung
Untersuchung des EDIT-Buffers, Neustart, Hilfsmenue-Aufruf

Hinweis: Die Lösungen zur Übung 1 befinden sich im Anhang A.

1. Rufen Sie das EDITOR-Programm von der Kommando-Ebene des Betriebssystems MAT 85 aus auf (siehe auch 3.):

a)

KMD> E

 <CR> oder <SPACE>

bzw.

b)

KMD> G(o)
START-ADR =5000

 <CR> oder <SPACE>
<CR> oder <SPACE>

2. Drücken Sie nach dem Erscheinen des EDITOR-Meldetextes die ESC-Taste.

3. Starten Sie den BFZ-EDITOR erneut, und rufen Sie das "Hilfs-Menue" auf (Taste "X"). Wurde das Hilfs-Menue installiert (siehe auch 1.2. und 3.), so erscheint jetzt die erste Seite des Merues auf dem Bildschirm. "Blättern" Sie das Hilfs-Menue durch (<SPACE>), und kehren Sie schließlich zum EDITOR-Melde-text zurück (<CR>).

4. Führen Sie einen "Kaltstart" durch (Taste "K"). Welche Start-adresse wird vorgeschlagen ?

START-ADR =

Ändern Sie die Startadresse auf 6200 und geben Sie <CR> oder <SPACE> ein. Wie lautet die Stopadresse ?

START-ADR =6000 6200

STOP -ADR =

Begründung:
.....

Geben Sie <CR> oder <SPACE> ein. Tippen Sie nach dem Löschen des Bildschirms folgenden Text ein:

DIESES IST EINE TESTZEILE.
AAA BBB CCC

 <CR>
<CR>

5. Geben Sie ein: CTRL-N (CTRL-Taste drücken und festhalten, anschließend "N" eingeben, CTRL-Taste loslassen). Führen Sie nach dem Erscheinen des EDITOR-Meldetextes einen "Warmstart" aus (<CR> oder <SPACE>). Wie lauten jetzt Start- und Stopadresse ?

START-ADR =

STOP -ADR =

Begründung:
.....

6. Nach der Rückkehr zum Monitorprogramm MAT-85 (Neustart: CTRL-N, anschließend ESC-Taste) können Sie den Inhalt des EDIT-Buffers ab Adresse 6200 mit Hilfe des PRINT-Modus darstellen (FORMAT = A).

```
KMD> P                <CR> oder <SPACE>
  START-ADR =6200
  STOP -ADR =6230
  FORMAT   =A
```

Durch welche Steuer-Bytes werden die einzelnen Zeilen des eingegebenen Textes getrennt ?

Zeilentrennung:

Welches Markierungs-Byte steht am Ende des EDIT-Buffers an der Stopadresse ?

Ende-Markierung:

7. Starten Sie den BFZ-EDITOR erneut und führen Sie einen Warmstart mit der vorgeschlagenen Adresse 6200 durch.

Wirkung:

Anschließend mit CTRL-N wieder zurück zum EDITOR-Meldetext und Kaltstart mit Adresse 6200.

Wirkung:

Nochmals zurück mit CTRL-N und Warmstart mit Adresse 6200.

Wirkung:
.....

8. Sehen Sie sich nach der Rückkehr nach MAT-85 den Inhalt des EDIT-Buffers an (wie unter 6.)

6200

Ändern Sie mit Hilfe des MEMORY-Modus (FORMAT = H) das 00-Byte unter der Adresse 6202 um in 20:

```
KMD> M                <CR>
  START-ADR =0000 6202  <CR>
  FORMAT    =A H       <CR>

6202 00 20           <CR>
```

Führen Sie nun einen Warmstart mit der EDIT-Buffer-Start-Adresse 6200 aus.

Wirkung:

.....

9. Schalten Sie das MFA-System aus und nach ca. 5 Sekunden wieder ein. Führen Sie jetzt einen Kaltstart des BFZ-EDITORS durch. Wie lautet die angebotene Startadresse ?

START-ADR =

Begründung:

.....

Übung 2:

Cursor-Steuerung und Carriage-Return:

1. Geben Sie nach einem Kaltstart des BFZ-EDITORS folgende Zeichen in den EDIT-Buffer ein:

```
Zeile 1: sovieler "A", bis Klingelzeichen ertönt,
         anschließend <CR>
Zeile 2: 5 x <SPACE> , 3 x "B" , 5 x <SPACE> , <CR>
Zeile 3: <CR>
Zeile 4: <CR>
Zeile 5: 3 x "C" , 5 x <SPACE> , kein <CR>
```

2. Bringen Sie den Cursor mit Hilfe der "Pfeil"-Tasten auf alle möglichen Positionen des erstellten Textes. Bewegen Sie den Cursor zum Schluß auf das zweite "C" der letzten Zeile, und geben Sie ein <CR> ein. Untersuchen Sie anschließend die Cursor-Bewegung in den Zeilen 5 und 6.

4.4. Zeichen radieren , Zeichen links löschen

Durch die Eingabe von

CTRL-R

(R = Radieren)

läßt sich das Text-Zeichen, auf das der Cursor positioniert ist, aus dem bestehenden Text entfernen. Um die Wirkung dieses Steuerbefehls auf dem Bildschirm sichtbar zu machen, wird das entsprechende Zeichen auf dem Bildschirm gelöscht und der um ein Zeichen nach links verschobene Zeilenrest erneut dargestellt. Je nach Länge des Zeilenrests und der Daten-Übertragungsrate zur Datensichtstation (Video-Interface) wird für die Ausgabe des Zeilenrests eine bestimmte Zeit benötigt. So ergibt sich z.B. für die Ausgabe einer Zeile mit 60 Zeichen bei einer Übertragungsrate von 1200 Bd (MFA-Video-Interface) eine Zeit von:

$$(60 \text{ Zeichen} \times 11 \text{ Bit/Zeichen}) / 1200 \text{ Bit/s} = 0,55 \text{ s}$$

11 Bit: 1 Start-Bit, 7 Daten-Bits, 1 Paritäts-Bit, 2 Stop-Bits

Für den Anwender bedeutet dieses, daß er beim Radieren eines oder mehrerer Zeichen die Bildschirmausgabe beobachten muß, um seine Tastatureingabe der Bildschirmausgabe anzupassen.

Zeilentrennungs-Zeichen "<" lassen sich durch "CTRL-R" nicht radieren. Hier muß entweder die gesamte Zeile gelöscht, bzw. die nachfolgende Zeile "angehängt" werden (siehe 4.6.). Beim Versuch, ein Zeilentrennungs-Zeichen zu radieren, ertönt als Bedienerhinweis ein Klingelzeichen.

Durch Drücken der

DEL-Taste

(Delete = Löschen)

läßt sich das Zeichen direkt links neben der Cursor-Position löschen. Auch hier wird nach dem Löschen des Textzeichens der um ein Zeichen nach links verschobene Zeilenrest erneut auf dem Bildschirm dargestellt (siehe oben "CTRL-R"). Wird mit "DEL" der linke Bildschirmrand erreicht, so ertönt als Bedienerhinweis ein Klingelzeichen. Die "DEL"-Funktion wird häufig bei der Eingabe eines Textes benötigt, um ein falsch eingegebenes Zeichen wieder zu löschen und durch das korrekte Zeichen zu ersetzen.

4.5. Einfüge-Modus Ein-/Aus

Wird beim BFZ-EDITOR ein Kaltstart ausgeführt, so befindet sich dieser während der Textbearbeitung im sogenannten "Einfüge-Modus". In dieser Betriebsart wird ein eingegebenes Textzeichen an der Cursor-Position in den Text eingeschoben und der Cursor um eine Stelle nach rechts verschoben. Es lassen sich auf diese Weise einzelne Zeichen oder Wörter auf einfache Art in den schon vorhandenen Text einfügen. Wird im Einfüge-Modus die maximale Zeilenlänge erreicht, so können keine weiteren Zeichen eingeschoben werden, und es ertönt ein Klingelzeichen als Bedienerhinweis.

Mit Hilfe des Einfüge-Modus lassen sich also auch Zeilen, die mit dem Zeilentrennungs-Zeichen "<" abgeschlossen wurden, auf die maximale Zeichenanzahl verlängern. Das Zeilentrennungs-Zeichen "<" selbst läßt sich nicht einfügen. Hierzu dient als besondere Funktion das "Einfügen einer Zeile" (siehe 4.6.).

Der "Einfüge-Modus" wird ausgeschaltet durch die Eingabe von

CTRL-E

(E = Einfügen)

Der BFZ-EDITOR befindet sich damit im sogenannten "Überschreib-Modus". Wird der Cursor in dieser Betriebsart auf ein "altes" Textzeichen positioniert und ein "neues" Zeichen eingegeben, so wird das "alte" durch das "neue" Zeichen überschrieben.

Durch erneute Eingabe von "CTRL-E" läßt sich der EDITOR wieder auf die Betriebsart "Einfügen" umschalten, und es ertönt als Hinweis ein Klingelzeichen. Der Bediener hat hierdurch eine Kontrolle über die aktuelle Betriebsart:

CTRL-E und	Klingelzeichen :	--> Einfüge-Modus EIN
CTRL-E und	kein Klingelzeichen :	--> Einfüge-Modus AUS

Bei einem Warmstart bleibt die zuletzt eingestellte Betriebsart erhalten.

4.6. Zeile löschen, Zeile einfügen, Zeile anhängen

Durch die Eingabe von

CTRL-ZL

(ZL = Zeile löschen)

(CTRL-Taste drücken und festhalten, "Z" eingeben, anschließend "L" eingeben, CTRL-Taste loslassen)

wird die Zeile, in der sich der Cursor befindet, aus dem Text entfernt. Um die Bildschirmanzeige zu aktualisieren, wird der "Bildschirmrest" ab der Cursor-Position erneut ausgegeben. Die dafür benötigte Zeit hängt von der Zeichenanzahl des "Bildschirmrests", sowie von der Daten-Übertragungsrate zur Datensichtstation ab (siehe auch 4.4.: "Zeichen radieren").

Mit Hilfe der Eingabe

CTRL-ZE

(ZE = Zeile einfügen)

wird an der Stelle der Cursor-Position ein Zeilentrennungs-Zeichen "<" eingefügt. Befindet sich der Cursor am Zeilenanfang, wird auf diese Weise eine neue Zeile, die nur aus dem Zeilentrennungs-Zeichen "<" besteht, vor der aktuellen Zeile eingefügt und der neue "Bildschirmrest" ausgegeben. Die Cursor-Position auf dem Bildschirm bleibt erhalten, er zeigt nunmehr auf den Anfang der neuen Zeile. Um diese neue Zeile beschreiben zu können, muß der Einfüge-Modus eingeschaltet sein (siehe 4.5.).

Steht der Cursor bei der Eingabe von "CTRL-ZE" nicht am Anfang einer Textzeile, so wird das Zeilentrennungs-Zeichen genau in dieser Cursor-Position eingefügt und der "Bildschirmrest" neu aufgebaut. Auf diese Weise lassen sich Textzeilen an der Cursor-Position trennen. Der Cursor wird anschließend auf das Ende der alten Zeile positioniert.

Ein Zusammenbinden von nacheinander liegenden Textzeilen ist möglich durch die Eingabe von

CTRL-ZA

(ZA = Zeile anhängen)

Dabei wird das Zeilentrennungs-Zeichen "<" aus der Zeile gelöscht, in der sich der Cursor befindet. Das hat die Wirkung, daß die nachfolgende Zeile an die aktuelle Zeile angehängt wird. Der Cursor wird dabei auf die "Verbindungsstelle" zwischen alter und angehängter Zeile geführt.

Wird beim Anhängen die maximale Zeichenanzahl für die neu entstehende Zeile überschritten, so wird diese automatisch auf die maximale Länge begrenzt, und der Rest der Zeile "rutscht" eine Zeile tiefer. Nach dem Anhängen einer Zeile wird der Bildschirminhalt aktualisiert.

Sollte nach dem Trennen einer Zeile mit CTRL-ZE der Bedarf bestehen, doch wieder die alte Zeile herzustellen, so läßt sich die Trennung durch die anschließende Eingabe von CTRL-ZA wieder rückgängig machen. Ebenso kann eine angehängte Zeile (CTRL-ZA) mit Hilfe einer nachfolgenden Trennung (CTRL-ZE) sofort wieder an ihre alte Position gebracht werden.

Übung 3

Zeichen löschen und einfügen,
Zeilen löschen, einfügen, trennen und anhängen

1. Geben Sie nach einem Kaltstart des EDITORs folgenden Text ein:

```
MIT HILFE DES EDITOR-<
<
PROGRAMMS LASSEN SICH      TEXTE ERSTELLEN <
<
      UND AUF DISKETTE ODER CASSETTE SPEICHERN. <
```

2. Ändern Sie den eingegebenen Text um in:

```
MIT HILFE DES BFZ-EDITORS LASSEN SICH TEXTE <
ERSTELLEN, BEARBEITEN UND AUF DISKETTE ODER <
CASSETTE SPEICHERN.<
```

Verwenden Sie dazu die Steuerfunktionen:

Cursor-Positionierung:	"Pfeil"-Tasten
Radieren einzelner Zeichen	(CTRL-R)
Ein- und Ausschalten des Einfüge-Modus	(CTRL-E)
Zeile löschen	(CTRL-ZL)
Zeile anhängen	(CTRL-ZA)
Zeile einfügen, bzw. Zeile trennen	(CTRL-ZE)

3. Ergänzen Sie den Text wie folgt:

```
AUFGABE DES EDITOR-PROGRAMMS: <
<
MIT HILFE DES BFZ-EDITORS LASSEN SICH TEXTE <
ERSTELLEN, BEARBEITEN UND AUF DISKETTE ODER <
CASSETTE SPEICHERN.<
```

4.7. Erweiterte Cursor-Bewegungen

Die erweiterten Cursor-Bewegungen dienen zum schnelleren Positionieren des Cursors innerhalb einer dargestellten Bildschirmseite:

- CTRL-CO (CO = Cursor nach oben)
Cursor-Positionierung auf die oberste Zeile
- CTRL-CU (CU = Cursor nach unten)
Cursor-Positionierung auf die unterste Zeile
- CTRL-CL (CL = Cursor nach links)
Cursor-Positionierung auf den Zeilenanfang
- CTRL-CR (CR = Cursor nach rechts)
Cursor-Positionierung auf das Zeilenende
- CTRL-W (W = nächstes Wort)
Cursor-Positionierung auf das nächste Wort

4.8. Tabulator

Mit Hilfe der Eingabe

CTRL-T (T = Tabulator)

wird der Cursor innerhalb einer Zeile auf die nächste Tabulatorposition bewegt. Die Tabulatorpositionen liegen jeweils 8 Zeichen voneinander entfernt, beginnend in der Spalte 0:

0	8	16	24	32	40	48	56	61
:	:	:	:	:	:	:	:	:
↑	↑	↑	↑	↑	↑	↑	↑	↑
								Zeilen- ende

Mit Hilfe der Tabulator-Funktion läßt sich der zu erstellende Text auf einfache Weise in tabellarischer Form anordnen, hier am Beispiel einer Stückliste gezeigt:

Beispiel:	Stückliste	
Kennz.	Benennung/Daten	Bemerkung
R1 ... R9	Widerstand 4k7	
R10... R17	Widerstand 560	
R18... R25	Widerstand 4k7	
C1, C2	Tantal-Elko 4,7 uF/25 V	Tropfenform
IC 1, IC 6	Sechs Inverter 74 LS 04	
IC 3, IC 4	4-Bit-Vergleicher 74LS85	
IC 5	8-Bit-D-Register 74LS273	
IC 2, IC 7	Sechs Inverter 7406	offener Kollektor
↑	↑	↑
0	16	48

Die Tabulator-Funktion kann im "Überschreib-Modus" oder auch im "Einfüge-Modus" benutzt werden:

Im "Überschreib-Modus" bleibt beim Einsatz des Tabulators der alte Text erhalten, der Cursor wird nur auf die nächste Tabulator-Spalte positioniert, z.B.:

```
L1: CALL TABELLE           Cursor Spalte 0
↑
                               Eingabe: CTRL-T

L1: CALL TABELLE           Cursor Spalte 8
↑
```

Erreicht der Cursor die maximale Zeilenlänge, so ertönt als Bedienerhinweis das Klingelzeichen.

Im "Einfüge-Modus" läßt sich mit Hilfe der erweiterten Cursor-Bewegung "CTRL-W" (Cursor auf das nächste Wort) und der Tabulator-Eingabe "CTRL-T" ein schon erstellter Text spaltenweise formatieren:

```
L1: CALL TABELLE           Cursor in Spalte 0
↑
                               Eingabe: CTRL-W ==>

L1: CALL TABELLE           Cursor nächstes Wort (Spalte 4)
↑
                               Eingabe: CTRL-T ==>

L1:   CALL TABELLE         Cursor mit Zeilenrest in Spalte 8
      ↑
                               Eingabe: CTRL-W ==>

L1:   CALL TABELLE         Cursor nächstes Wort (Spalte 13)
      ↑
                               Eingabe: CTRL-T ==>

L1:   CALL   TABELLE       Cursor mit Zeilenrest Spalte 16
      :       ↑
      :       :
      0       8       16
```

Wird durch das Einfügen von Leerzeichen mit Hilfe des Tabulators die maximale Zeilenlänge erreicht, so kann der Cursor mit dem rechten Zeilenrest unter Umständen nicht die nächste Tabulator-Spalte erreichen. Die Cursorbewegung wird dann abgebrochen und ein akustisches Signal als Meldung für den Bediener ausgegeben.

Beispiel:

```
L1:      CALL      TABELLE ;Aufruf des Unterprogramms TABELLE
          ↑
          :
          Cursor                               Zeilenende
```

Eingabe: CTRL-T

```
L1:      CALL      TABELLE ;Aufruf des Unterprogramms TABELLE
          ↑      ↑
          :      :
          Cursor  :
          (Ist-Position) nächste Tabulator-Spalte
                               (Soll-Position)
```

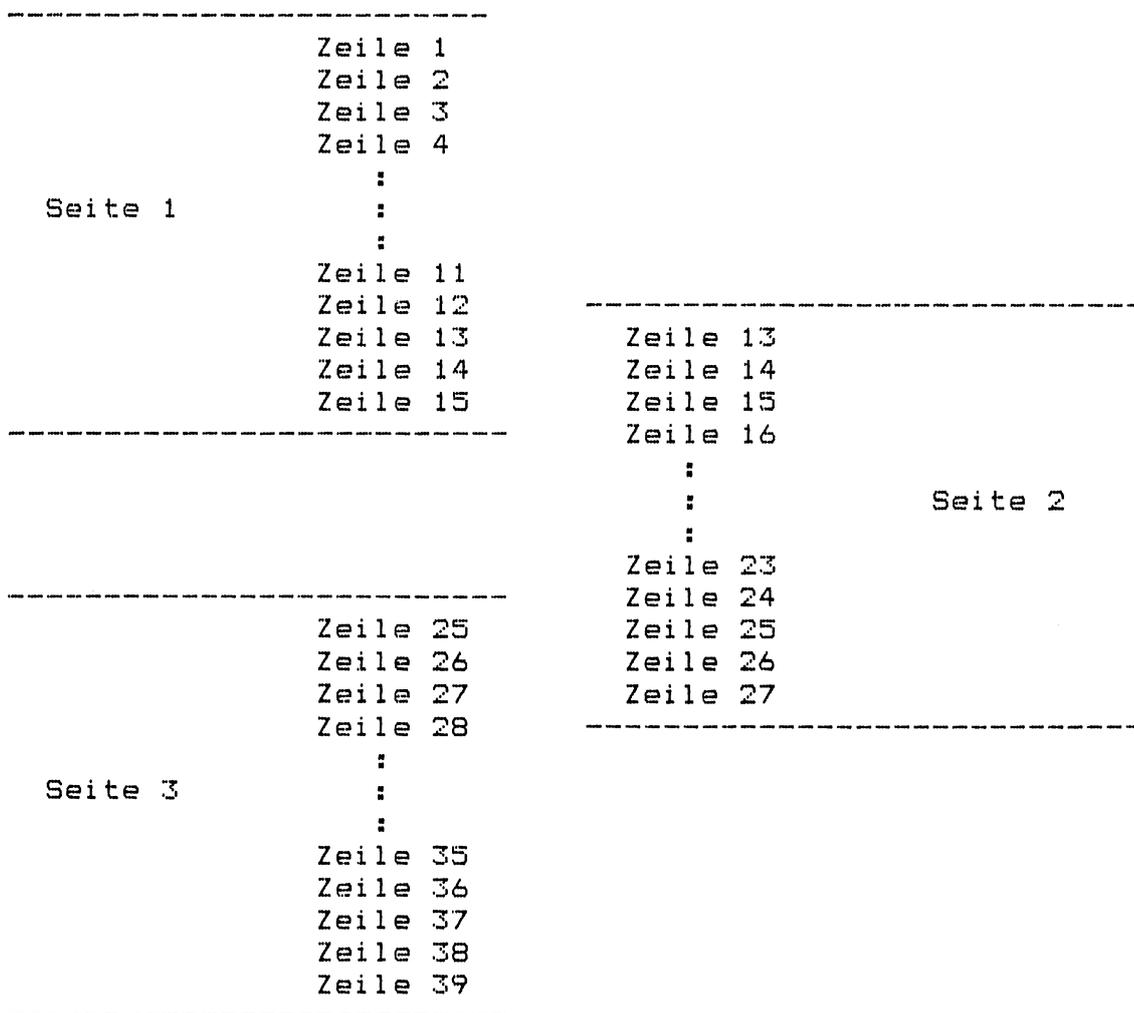
4.9. Seiten blättern

Mit Hilfe folgender EDITOR-Steuerzeichen läßt sich ein längerer Text seitenweise auf dem Bildschirm darstellen:

- CTRL-O eine Seite nach "oben" blättern
- CTRL-U eine Seite nach "unten" blättern
- CTRL-CA Cursor zum Textanfang
(Ausgabe der ersten Textseite, Cursor an den Textanfang)
- CTRL-CE Cursor zum Textende
(Ausgabe der letzten Textseite, Cursor an das Textende)

Das Blättern nach "oben" (bzw. nach "unten") wird nicht ausgeführt, wenn auf dem Bildschirm die erste (bzw. die letzte) Textseite dargestellt ist (Klingelzeichen als Bedienerhinweis).

Beim seitenweisen Blättern "überlappen" sich die dargestellten Textseiten um jeweils drei Zeilen. Dadurch bleibt der inhaltliche Zusammenhang des Textes sichtbar:



4.10. Finden eines "Text-Strings", Ersetzen eines "Text-Strings"

Der sogenannte "FIND"-Modus dient zum Auffinden von Zeichenketten ("Strings") innerhalb des bearbeiteten Textes. Je nach Bedarf kann die gesuchte Zeichenkette durch eine neue ersetzt werden.

Durch die Eingabe von

CTRL-FS (FS = "FIND"-String setzen)

wird der Bildschirm gelöscht, und in der linken oberen Bildschirmcke erscheint der Meldetext:

FINDE : <

Der Anwender hat nun die Möglichkeit, eine beliebige Zeichenkette mit einer maximalen Länge von 31 Zeichen als "Find"-String einzugeben (auch Leerzeichen, Ziffern oder Satzzeichen). Eine Korrektur der Eingabe ist mit Hilfe der "DEL"-Taste möglich. Mit Hilfe von <CR> wird die Definition des "Find"-String abgeschlossen.

Durch die Ausgabe des Meldetextes

ERSETZE: <

erfolgt die Aufforderung an den Bediener, bei Bedarf eine "Ersatz"-Zeichenkette einzugeben (maximal 31 Zeichen, Eingabekorrektur wiederum mit Hilfe der "DEL"-Taste). Die Eingabe wird mit <CR> beendet.

Wird "CTRL-FS" zu einem späteren Zeitpunkt erneut aufgerufen, so erfolgt nach der Ausgabe des Meldetextes die Darstellung des alten "Find"- bzw. "Ersatz"-String. Diese einmal definierten Zeichenketten bleiben erhalten, wenn sie durch die Eingabe von <CR> vom Anwender quittiert werden. Wird ein anderes Zeichen als <CR> angegeben, können jeweils neue Zeichenketten eingegeben werden.

Beispiel:

- a) **FINDE : SUCHWORT-ALT<** **<CR>** : SUCHWORT-ALT bleibt erhalten
- b) **FINDE : N** **N** : neues Suchwort wird definiert, N ist erstes neues Zeichen
- FINDE : Neues Suchwort<** **eues Suchwort <CR>**

Sind "Find"- und auch "Ersatz"-String definiert worden, so erfolgt anschließend die Suche nach der gewünschten Zeichenkette innerhalb des EDIT-Buffers, und zwar ab der aktuellen Cursor-Position.

Dabei bleiben Groß- oder Kleinschreibung innerhalb des "Find"-String unberücksichtigt. D.h., mit dem Suchwort

"ERGEBNIS"

wird z.B. auch gefunden:

"Ergebnis"
"Zwischenergebnis"
"ergebnislos"

War die Suche erfolgreich, so wird die gesuchte Bildschirmseite dargestellt. Die gefundene Zeichenkette befindet sich in der obersten Bildschirmzeile, der Cursor zeigt auf den Anfang dieser Zeichenkette. Wurde die gesuchte Zeichenkette nicht gefunden, so wird der "alte" Bildschirminhalt wieder dargestellt, und es ertönt das Klingelzeichen.

Durch die Eingabe von

CTRL-FE

(FE = "Find"-String ersetzen)

läßt sich die gefundene Zeichenkette durch die definierte "Ersatz"-Zeichenkette austauschen. Um Fehlbedienungen zu vermeiden, ist ein Austausch nur dann möglich, wenn der Cursor bei der Eingabe von "CTRL-FE" auf den Anfang des Such-String positioniert ist. Ist der "Ersatz"-String länger als der "Such"-String, so kann die neu entstehende Zeile unter Umständen die maximale Zeilenlänge überschreiten. In diesem Fall wird die Zeile auf ihre maximale Länge beschnitten, der Zeilenrest "rutscht" dann eine Zeile tiefer. Ist der Ersatz-String "leer", so wird der "Such-string" beim Ersetzen lediglich aus dem Text gelöscht.

Mit Hilfe von

CTRL-FF

(FF = "Find"-String finden)

läßt sich jederzeit eine einmal definierte Zeichenkette erneut im EDIT-Buffer suchen und je nach Bedarf austauschen. Wird das gesuchte Wort nicht gefunden, so ertönt als Bedienerhinweis das Klingelzeichen.

Bei einem Warmstart des EDITOR-Programms bleiben einmal definierte Such- oder Ersatz-Zeichenketten erhalten. Wird ein Kaltstart ausgeführt, so werden die "alten" Zeichenketten gelöscht.

4.11. Block-Befehle

4.11.1 Block markieren, Block-Markierungen rücksetzen, Block-Markierungen finden

Mit Hilfe des "BLOCK"-Modus lassen sich beliebige zusammenhängende Text-Blöcke markieren, löschen, kopieren, verschieben oder aus dem EDIT-Buffer heraus als Text-Datei auf die Diskette oder Cassette speichern.

Das Markieren von Text-Blöcken ist nur zeilenweise möglich. Dazu wird der Cursor auf die erste Zeile positioniert, die noch zu dem Text-Block gehören soll (siehe Beispiel unten: 1 -->). Durch die Eingabe von

CTRL-BM

(BM = Block markieren)

wird die sogenannte Block-Marke "--B--" vor den zu markierenden Text-Block gesetzt.

Das Ende des Text-Blocks wird auf die gleiche Art festgelegt: der Cursor wird auf die erste Zeile positioniert, die nicht mehr zu dem Text-Block gehören soll (siehe Beispiel unten: 2 -->). Mit "CTRL-BM" wird die Block-Marke "--B--" nun direkt hinter dem zu markierenden Text-Block eingefügt. Dieser ist nun also durch zwei Block-Marken eingeschlossen:

```

1 --> Diese Zeile gehört noch nicht zu dem Text-Block.<
      --B--<
      Hier beginnt der markierte Text.<
      Die Länge des markierten Textes<
      ist beliebig.<
      Das ist die letzte Zeile des Text-Blocks.<
      --B--<
2 --> Diese Zeile gehört nicht mehr zu dem Text-Block.<

```

Die Zeichenkette "--B--" wird vom EDITOR-Programm nur dann als Blockmarke gewertet, wenn sie völlig allein in einer Zeile steht. Werden vom Anwender in der markierten Zeile zusätzliche Zeichen vor oder hinter die Block-Marke geschrieben, so verliert diese Block-Markierung ihre Funktion (Sicherung vor Fehlbedienung).

Alle im Text befindlichen Block-Marken "--B--" lassen sich durch die Eingabe von

CTRL-BR

(BR = Block-Markierung
rücksetzen)

zurücksetzen.

Mit Hilfe von

CTRL-BF

(BF = Block-Markierung
finden)

werden Block-Marken "--B--" ab der aktuellen Cursor-Position gesucht und die gefundene Seite auf dem Bildschirm dargestellt.

4.11.2 Block löschen, Block kopieren, Block verschieben

Die nachfolgenden Block-Befehle dienen dazu, einen markierten Block weiter zu verarbeiten. Ein Text-Block kann nur dann eindeutig in seinem Anfang und seinem Ende definiert sein, wenn innerhalb des gesamten Textes genau zwei Block-Marken "--B--" gesetzt sind. Ist dieses nicht der Fall (z.B. keine Block-Marke oder nur eine oder mehr als zwei Block-Marken), so erfolgt beim Aufruf aller folgenden Block-Befehle die Fehlermeldung:

```

--- FEHLER: BLOCK-MARKIERUNG
==> <CR>,<SPACE>

```

Durch die Eingabe von

```
CTRL-BL
```

(BL = Block löschen)

wird ein markierter Block (inklusive seiner beiden Block-Marken) aus dem EDIT-Buffer gelöscht. Dieser EDITOR-Steuerbefehl ist mit Vorsicht zu verwenden, da der gelöschte Text-Block in den meisten Fällen unwiederbringlich aus dem Arbeitsspeicher des Computers entfernt wird.

Beispiel:

Vor CTRL-BL:

```

Dieser Text gehört nicht zum markierten <
Block und wird nicht gelöscht.<
--B--<
Markierter Block<
--B--<
Ab hier soll nicht mehr gelöscht werden.<

```

Nach CTRL-BL:

```

Dieser Text gehört nicht zum markierten <
Block und wird nicht gelöscht.<
Ab hier soll nicht mehr gelöscht werden.<

```

Mit Hilfe von

CTRL-BK

(BK = Block kopieren)

läßt sich ein markierter Text-Block an eine beliebige Stelle innerhalb des Textes duplizieren. Der kopierte Block wird dabei vor die Zeile der aktuellen Cursor-Position eingeschoben. Die Block-Marken des "alten" Blocks werden gelöscht, der kopierte Block ist erneut durch die beiden Block-Marken gekennzeichnet:

Beispiel:

Vor CTRL-BK:

Cursor -->

```
Dieser Text wird nicht kopiert.<
--B--<
Markierter Block<
--B--<
Diese Zeile wird ebenfalls nicht kopiert.<
```

Nach CTRL-BK:

Cursor -->

```
--B--<
Markierter Block<
--B--<
Dieser Text wird nicht kopiert.<
Markierter Block<
Diese Zeile wird ebenfalls nicht kopiert.<
```

Hinweise:

1. Der Kopierversuch eines Blocks in die eigenen Block-Grenzen wird durch das EDITOR-Programm abgefangen, und es ertönt ein Klingelzeichen als Fehlermeldung.
2. Je nach Länge des gesamten EDIT-Buffers und Länge des zu kopierenden Blocks wird für das Duplizieren des Text-Blockes eine bestimmte Zeit benötigt. Diese Wartezeit macht sich allerdings erst bei Buffer-Längen von einigen 10 KByte bemerkbar.

Der Block-Befehl

CTRL-BV

(BV = Block verschieben)

dient dazu, einen markierten Textblock (einschließlich der Block-Markierung "--B--") innerhalb des EDIT-Buffers zu verschieben. Der verschobene Block wird dabei vor die Zeile der aktuellen Cursor-Position eingesetzt.

Beispiel:

Vor CTRL-BV:

Cursor -->

```

Dieser Text wird nicht verschoben.<
--B--<
Markierter Block<
--B--<
Der markierte Block soll <
hinter diese Zeile verschoben werden.<
In dieser Zeile muß der Cursor stehen.<

```

Nach CTRL-BV:

Cursor -->

```

Dieser Text wird nicht verschoben.<
Der markierte Block soll <
hinter diese Zeile verschoben werden.<
--B--<
Markierter Block<
--B--<
In dieser Zeile muß der Cursor stehen.<

```

Hinweise:

1. Auch hier gilt, daß das Verschieben eines Blocks in die eigenen Block-Grenzen nicht möglich ist (Klingelzeichen).
2. Um einen Text-Block innerhalb des EDIT-Buffers zu verschieben, lassen sich verschieden Verfahren anwenden, z.B.:
 - a) Der gesamte Block wird auf einmal verschoben, d.h. zuerst erfolgt ein Duplizieren des "Quellblocks" an die gewünschte Stelle, anschließend wird der "Quellblock" gelöscht. Diese Verschiebetechnik ist verhältnismäßig schnell, benötigt aber unter Umständen je nach Länge des zu verschiebenden Blocks zusätzlichen Speicherplatz (nach dem Kopieren des "Quellblocks" ist dieser zweimal innerhalb des EDIT-Buffers enthalten).
 - b) Der Block wird Zeichen für Zeichen verschoben, d.h. nach dem Kopieren eines Zeichens des Quellblocks in den Zielbereich wird dieses Zeichen erst aus dem Quellblock gelöscht, bevor das nächste Zeichen verschoben wird. Der EDIT-Buffer verlängert sich während des Verschiebevorgangs also nur immer um ein Zeichen, anschließend wird die alte EDIT-Buffer-Länge wieder hergestellt. Der Nachteil dieser Speicherplatz-sparenden Verschiebetechnik ist allerdings ein gehöriger Zeitaufwand (Platz machen für ein Zeichen im Zielbereich, kopieren des Zeichens, löschen des Zeichens aus dem Quellbereich).

Das im EDITOR-Programm verwendete Verschiebeverfahren stellt einen Kompromiß beider dargestellten Verfahren dar:

Das Verschieben erfolgt Zeile für Zeile, d.h. zuerst wird im Zielbereich des EDIT-Buffers Platz geschaffen für die zu kopierende Zeile, dann wird die Zeile aus dem Quellbereich in den Zielbereich dupliziert und anschließend kann die übertragene Zeile aus dem Quellbereich gelöscht werden. Dieser Vorgang wird für jede Zeile des zu verschiebenden Blocks durchgeführt.

Während des Verschiebevorgangs wird der EDIT-Buffer also maximal um die Länge einer Zeile vergrößert. Dadurch wird zum Verschieben eines Blocks kein wesentlich größerer Speicherplatz benötigt. Der Zeitbedarf für das "Auf- und Ab-Pumpen" des EDIT-Buffers (Kopieren und Löschen einer Zeile) hält sich in vertretbaren Grenzen und macht sich erst bei großen zu verschiebenden Textblöcken und EDIT-Buffer-Längen von einigen 10 KByte bemerkbar.

Ein markierter Block läßt sich mit Hilfe von

CTRL-BS

(BS = Block sichern)

auf die Diskette oder Cassette speichern. Dabei wird der Textblock ohne die Block-Markierungen "--B--" zum angeschlossenen Diskettenlaufwerk oder zum MFA-Cassetten-Interface übertragen. Der gesicherte Textblock wird dabei als eigene Textdatei abgespeichert und läßt sich später wieder in den EDIT-Buffer laden (siehe auch 4.12. "Speichern und Laden mit Hilfe von Diskette oder Cassette").

4.12. Speichern und Laden mit Hilfe von Diskette oder Cassette

4.12.1 Speichern auf Diskette oder Cassette

Mit Hilfe von

CTRL-S

(S = Speichern/Laden)

läßt sich der Inhalt des aktuellen EDIT-Buffers auf die Diskette oder die Cassette speichern.

Nach dem Aufruf der Steuerfunktion CTRL-S wird der Bildschirm gelöscht und folgender Meldetext ausgegeben:

```
--- SPEICHERN/LADEN  
  
F = FLOPPY  
C = CASSETTE  
E = EDITOR  
  
> -
```

Wie in 4.11.2 beschrieben, läßt sich auch ein markierter Block mit Hilfe des Block-Befehls "CTRL-BS" auf Diskette oder Cassette speichern. Zur Unterscheidung erfolgt beim "Block"-Speichern die veränderte Meldetext-Ausgabe:

```
--- BLOCK SPEICHERN  
  
F = FLOPPY  
C = CASSETTE  
E = EDITOR  
  
> -
```

Durch die Eingabe des Zeichens "F" erfolgt ein Aufruf des BFZ-MINI-DOS (DOS = Disk Operating System, Disketten-Verwaltungsprogramm). Ist das BFZ-MINI-DOS nicht installiert, so wird eine akustische Fehlermeldung ausgegeben, und obiger Meldetext erscheint erneut.

Mit Hilfe des BFZ-MINI-DOS läßt sich der Inhalt des EDIT-Buffers auf die Diskette speichern:

"SAVE"	S	<CR>	eingeben
"Laufwerk"	A oder B	<CR>	eingeben
"NAME:"	NAME.EXT		eingeben
			(NAME max. 8 Buchstaben EXT max. 3 Buchstaben)
			(EXT = engl. Extension, dtsch. Erweiterung)

(siehe auch "Fachpraktische Übung, Band 3: Floppy-Disk-Controller")

Die Übergabe der Start- und Stopadresse des zu sichernden Speicherbereiches an das BFZ-MINI-DOS erfolgt automatisch durch das EDITOR-Programm. Die ersten beiden Steuerbytes (0D 0A, siehe auch 3.), sowie die Buffer-Ende-Markierung (00) werden nicht auf die Diskette gespeichert. Bei dem Versuch, einen "leeren" EDIT-Buffer zu speichern, erfolgt die Fehlermeldung

*** FEHLER: PROGRAMM-SPEICHER LEER
==> SPACE_

Beim Speichern auf die Diskette wird die Textdatei im Disketten-Directory (Inhaltsverzeichnis) automatisch mit dem Typ "???" gekennzeichnet. Beim späteren Laden einer Textdatei von der Diskette lassen sich mit Hilfe des BFZ-EDITORS nur Dateien dieses Typs in den EDIT-Buffer übertragen. Damit ist gewährleistet, daß von der Diskette nur Textdateien, die mit dem EDITOR erstellt wurden, wieder in den EDIT-Buffer geladen werden können. Der Rücksprung zum BFZ-EDITOR erfolgt mit Hilfe von "Q" (Quit).

Beispiel:

Ein Text im EDIT-Buffer (START-ADR =6000, STOP -ADR =7200)
 soll auf die Diskette gespeichert werden.

Eingabe: CTRL-S ==> "Meldetext-Ausgabe:"

```

--- SPEICHERN/LADEN

F = FLOPPY
C = CASSETTE
E = EDITOR

> _
    
```

Eingabe: F ==> "Meldetext-Ausgabe des
 BFZ-MINI-DOS"

Eingabe: S ==> "SAVE"
 Eingabe: A ==> "LAUFWERK A"
 "NAME:"

Eingabe des Datei-Namen, z.B.:

TEST1.ASM ==> "Text-Datei wird auf
 die Diskette gespeichert"

 "Meldetext-Ausgabe des
 BFZ-MINI-DOS"

z.B. Eingabe: D ==> "DIRECTORY"
 Eingabe: A ==> "LAUFWERK A"

==> "Directory-Ausgabe des BFZ-MINI-DOS:"

---	NAME	---	TYP	START (HEX)	STOP (HEX)	BLOECKE
	TEST1.ASM		???	6002	71FF	2

Zum Speichern einer Textdatei auf die Cassette muß der Bediener nach der Ausgabe des oben angegebenen Meldetextes die Taste "C" betätigen. Der Meldetext wird daraufhin wie folgt erweitert:

```

--- CASSETTE

S = SAVE
L = LOAD
E = EDITOR

> _
```

Durch die Eingabe von "S" wird die SAVE-Routine für das MFA-Cassetten-Interface aufgerufen, und der EDIT-Buffer-Inhalt läßt sich als Textdatei auf die Cassette speichern (siehe auch "Fachtheoretische Übungen, Beschreibung MAT-85"). Auch hier erfolgt die Übergabe der Start- und Stop-Adressen an die Cassetten-SAVE-Routine automatisch durch das EDITOR-Programm.

4.12.2 Laden einer Textdatei von Diskette oder Cassette

Das Laden einer Textdatei von der Diskette oder der Cassette in den EDIT-Buffer erfolgt ebenfalls durch die Eingabe der Steuerfunktion

CTRL-S

(S = Speichern/Laden)

Wie schon in 4.12.1 beschrieben, erfolgt nach der Ausgabe des Meldetextes die Auswahl durch den Bediener ("F" = FLOPPY, "C" = CASSETTE).

Wurde das BFZ-MINI-DOS aufgerufen ("F" = FLOPPY), so läßt sich mit Hilfe von

"LOAD"	L	<CR> eingeben
"Laufwerkauswahl"	A oder B	<CR> eingeben
"NAME:"	NAME.EXT	eingeben

eine Textdatei (Directory-Typ: "???") von der Diskette in den Arbeitsspeicher des Computers übertragen. Dabei wird diese Textdatei immer an das Ende des aktuellen EDIT-Buffers geladen. Dadurch lassen sich Texte, die mit Hilfe des EDITORs auf die Diskette gespeichert worden sind, aneinanderbinden (engl.: "Merge", dtsh. "Vereinigung"). Ist der EDIT-Buffer "leer" (nach einem Kaltstart), so befindet sich nach dem Ladevorgang nur der von der Diskette übertragene Text im EDIT-Buffer.

Nach dem Rücksprung zum BFZ-EDITOR mit Hilfe von "Q" (Quit) steht der geladene Text zur weiteren Bearbeitung im EDIT-Buffer bereit.

Ebenso erfolgt die Übertragung einer Textdatei von der Cassette an das Ende des aktuellen EDIT-Buffers, wenn der Bediener nach der Auswahl von "C" (CASSETTE) die "LOAD"-Routine für das MFA-Cassetten-Interface aufruft.

4.13. Drucken des EDIT-Buffer-Inhalts

4.13.1 Drucker-Menue

Das Drucker-Menue wird aufgerufen mit Hilfe von

CTRL-D

(D = Drucker)

Der Bildschirm wird gelöscht, und es erscheint der Meldetext des Drucker-Menues:

```

--- DRUCKER-MENUE
ZEILENRAND:  8 _
  
```

Der vorgegebene Zeilenrand für den Drucker kann mit Hilfe von <CR> oder <SPACE> angenommen oder durch die Eingabe eines neuen Zeilenrandes (maximal 99 dez.) mit anschließendem <CR> oder <SPACE> vom Bediener verändert werden.

Mit der Ergänzung des Meldetextes

```

ZEILEN/SEITE (0 = ENDLOSDRUCK):  0 _
  
```

wird der Bediener aufgefordert, die maximale Zeilenzahl für eine Druck-Seite anzugeben (maximal 99 dez.). Wird beim Ausdrucken diese maximale Zeilenzahl erreicht, so erfolgt automatisch ein Formularvorschub auf das nächste Blatt. Die Länge des Formularvorschubs ist Drucker-spezifisch und muß daher am Drucker eingestellt werden. Wird eine "0" eingegeben, so erfolgt der Ausdruck ohne Überprüfung der maximalen Zeilenzahl pro Druck-Seite (Endlosdruck).

Mit Hilfe von

```

SEITEN-NUMERIERUNG:  1 _
  
```

wird der Ausdruck der ersten Seiten-Nummer (maximal 99 dez.) innerhalb einer "Kopfzeile" festgelegt (siehe weiter unten).

Nach der Vervollständigung des Meldetextes

```

D = DRUCKEN
E = EDITOR
> _
  
```

kann der Bediener das Ausdrucken des EDIT-Buffer-Inhalts durch die Eingabe von "D" einleiten.

Ist kein Drucker angeschlossen oder dieser nicht eingeschaltet, so erfolgt eine akustische und optische Fehlermeldung:

```
*** PRINTER NOT READY ***  
==> <CR>, <SPACE>
```

Bei angeschlossenem Drucker in Bereitschaftsstellung wird nach der Eingabe von "D" der Inhalt des EDIT-Buffers zum Drucker übertragen.

Hinweis: Das MFA-Video-Interface stellt alle verwendeten Buchstaben in Großschreibung dar. Im EDIT-Buffer werden aber alle Buchstaben so abgelegt, wie sie auch tatsächlich mit der Tastatur eingegeben wurden (z.B. Tastatur-Eingabe "a" ==> "a" wird als ASCII-Zeichen (= 61 hex) im EDIT-Buffer abgelegt ==> auf dem Bildschirm erscheint "A"). Beim Ausdrucken des EDIT-Buffer-Inhaltes werden aber alle als Kleinbuchstaben eingegebenen Zeichen auch als Kleinbuchstaben gedruckt, so daß der gedruckte Text u.U. bzgl. der Groß- und Kleinschreibung nicht mit dem auf dem Bildschirm dargestellten Text übereinstimmt. Soll auch die Ausgabe auf dem Drucker nur in Großschreibung erfolgen, so muß bei der Texteingabe die Taste

```
"ALPHA-LOCK"
```

aktiviert werden (Ausnahme: der Ausdruck von Programm-Listings beim Assemblieren erfolgt immer in Großschreibung, siehe 4.14).

4.13.2 Steuerung des Formularvorschubs mit ".", Ausdruck von Kopfzeilen

Wie unter 4.13.1 erläutert, läßt sich der Formularvorschub durch die Eingabe der maximalen Zeilenzahl pro Druck-Seite in festen Abständen einstellen. Eine weitere Möglichkeit zur Steuerung des Blatt-Vorschubs bieten die sogenannten "Punkt-Zeilen":

- Beginnt eine Zeile des Textes mit einem Punkt ".", so wird an dieser Stelle ein Formularvorschub beim Drucker ausgelöst. Der Rest der Zeile wird beim Ausdrucken ignoriert.

Eine zusätzliche Bedeutung haben "Punkt-Zeilen" am Anfang des Textes:

- Alle "Punkt-Zeilen", die direkt am Anfang des EDIT-Buffers liegen, werden als sogenannte Kopfzeilen definiert. Diese Kopfzeilen werden nach jedem Formularvorschub komplett ausgedruckt.
- Mit Hilfe des "Numerus"-Zeichens "#" läßt sich der Ausdruck einer fortlaufenden Seitenzahl innerhalb einer Kopfzeile festlegen. Der Beginn der Seiten-Numerierung kann mit Hilfe des Drucker-Menues eingestellt werden.

Beispiel:

<pre>.Dieses ist die 1. Zeile des EDIT-Buffers< .< . --- Seite# ---< . Hier beginnt die erste Druckseite.< Anschließend soll ein Formular-< vorschub erfolgen.< .< Der Text wird an dieser Stelle mit< der zweiten Druckseite fortgesetzt.< Anschließend soll ein Formular-< vorschub erfolgen.< .Der Rest der Zeile wird ignoriert.< Hier beginnt die dritte Druckseite.<</pre>	<pre>(Kopfzeile 1) (Kopfzeile 2) (Kopfzeile 3) (Kopfzeile 4) ← (Formularvorschub) ← (Formularvorschub)</pre>
---	--

Durch den Aufruf des Drucker-Menues wird z.B. gewählt:

```
ZEILENRAND:      8
ZEILEN/SEITE (Ø = ENDLOSDRUCK):  0
SEITEN-NUMERIERUNG:  1
```

Das Druck-Ergebnis hätte dann folgende Form:

Dieses ist die 1. Zeile des EDIT-Buffers

(Seite 1)

--- Seite 1 ---

Hier beginnt die erste Druckseite.
Anschließend soll ein Formular-
vorschub erfolgen.

Dieses ist die 1. Zeile des EDIT-Buffers

(Seite 2)

--- Seite 2 ---

Der Text wird an dieser Stelle mit
der zweiten Druckseite fortgesetzt.
Anschließend soll ein Formular-
vorschub erfolgen.

Dieses ist die 1. Zeile des EDIT-Buffers

(Seite 3)

--- Seite 3 ---

Hier beginnt die dritte Druckseite.

4.14. Assemblieren

Wie in der Einleitung schon beschrieben, eignet sich der BFZ-EDITOR besonders dazu, Assembler-Quell-Programme zu erstellen, zu bearbeiten und anschließend durch den Assembliervorgang in ein Maschinen-Programm umzusetzen. Die Bearbeitung des Assembler-Quelltextes erfolgt mit Hilfe des EDITORS im EDIT-Buffer (Vorschlags-Start-Adresse z.B.: 6000 hex.). Je nach Länge des erstellten Assembler-Programms verschiebt sich die EDIT-Buffer-Stop-Adresse entsprechend. Dabei ist zu beachten, daß ein voll kommentierter Assembler-Text von etwa einer DIN-A4-Seite (das entspricht ca. 60 Zeilen, bzw. ca. 4 Bildschirmseiten) einen EDIT-Buffer von ca. 1,5 bis 2 KByte Länge benötigt (d.h., die EDIT-Buffer-Stop-Adresse liegt dann bei ca. 6600 bis 6800 hex.).

Durch den Assembliervorgang läßt sich der erstellte Quell-Text in ein zugehöriges Maschinen-Programm umsetzen (Vorschlags-Adresse z.B.: E000 hex.). Für ein 60-zeiliges Maschinen-Programm mit durchschnittlich 2 Byte Maschinen-Code pro Zeile ergibt sich somit ein Speicherbedarf von ca. 120 Byte (d.h., E000 bis ca. E078 hex.).

Für den Assemblier-Vorgang wird der "Line-By-Line-Assembler" des Betriebssystems MAT 85 benutzt. Aus diesem Grund muß der erstellte Assembler-Text den Regeln des MAT 85 - Assemblers genügen (siehe MAT 85 - Beschreibung).

Ein mit dem EDITOR erstelltes Programm im 8085-Assembler-Code kann assembliert werden durch die Eingabe von

CTRL-A

(A = Assemblieren)

Nach dieser Eingabe wird der Bildschirm gelöscht, und es erscheint folgender Meldetext:

```

--- ASSEMBLIEREN

1 = MIT BILDSCHIRM-LISTING
2 = OHNE BILDSCHIRM-LISTING
3 = MIT DRUCKER-LISTING
E = EDITOR

> _

```

Mit der Eingabe der Ziffern 1, 2 oder 3 wird der Assemblier-Vorgang gestartet. Dabei ist folgendes zu beachten:

- a) Die "Start-Adresse" für den erzeugten Maschinen-Code wird bei jedem Assemblier-Vorgang auf "0000" eingestellt (EPROM-Bereich des Betriebs-Systems, daher kein unbeabsichtigtes Überschreiben eines wichtigen RAM-Bereichs möglich). Die eigentliche Wahl der Start-Adresse des erzeugten Maschinen-Programms erfolgt mit Hilfe der "ORG"-Anweisung, z.B.:

```

ORG 0E000 ;Programm-Anfang bei 0E000

```

Bei der Wahl der Anfangsadresse des Maschinenprogramms mit Hilfe der "ORG"-Anweisung ist unbedingt darauf zu achten, daß durch den erzeugten Maschinencode kein wichtiger RAM-Bereich (z.B. der Quelltext im EDIT-Buffer) überschrieben wird.

- b) Vor jedem Assemblier-Vorgang wird die "Label-Tabelle" des MAT 85 - Assemblers gelöscht. Alle im Assembler-Text verwendeten Labels werden also bei einem Assemblier-Vorgang neu erzeugt. Für die Verwendung von Labels gilt folgende wichtige Vereinbarung:

!!!	Label-Definitionen müssen immer in der ersten	!!!
!!!	Spalte einer Programm-Zeile stehen	!!!
!!!	Alle anderen Assembler-Kommandos dürfen erst ab	!!!
!!!	der zweiten Spalte einer Programm-Zeile stehen	!!!
!!!	Kommentare (Kennzeichnung durch vorangestelltes ";"	!!!
!!!	dürfen in jeder Spalte einer Programm-Zeile beginnen	!!!

Label-Definitionen können auf zwei Arten vorgenommen werden:

- i) Definition symbolischer Adressen mit dem Doppelpunkt, z.B.

```
ANF:
NEXT:
LOOP:
```

- ii) Definition von Hex-Konstanten, bzw. festliegenden Adressen mit Hilfe der "EQU"-Anweisung, z.B.

```
EING    EQU    01    ;8-BIT-PARALLEL-EINGABE
AUSG    EQU    02    ;8-BIT-PARALLEL-AUSGABE
TAST    EQU    0043  ;TASTATUR-ABFRAGE
KONST   EQU    55    ;KONSTANTE = 55H
```

Die Definitionen mit Hilfe der "EQU"-Anweisung müssen am Anfang eines Assembler-Programms vorgenommen werden. Dadurch hat der Anwender die Möglichkeit, diese im Bedarfsfall einfach und schnell im gesamten Programm abzuändern.

Wird die beschriebene Vereinbarung bei der Definition von Labels nicht eingehalten, so führt dieses beim Assemblieren mit Bildschirm-Listing zu einer Fehlermeldung. Aus diesem Grund empfiehlt es sich, ein Assembler-Programm bei der Eingabe mit Hilfe der "Tabulator"-Funktion zu formatieren. Dieses erhöht außerdem die Übersichtlichkeit des erstellten Quelltextes (siehe Beispiel am Ende des Kapitels).

Zur Kontrolle, ob alle' verwendeten Labels auch definiert wurden, sollte am Ende des Assembler-Textes die Anweisung

LP

stehen ("Label-Print"). Bei der Darstellung der Label-Tabelle werden dann alle nicht definierten Labels mit einem "*" gekennzeichnet.

- c) Die "END"-Anweisung zur Beendigung der Programm-Eingabe ist beim Assemblieren mit Hilfe des BFZ-EDITORS überflüssig, da das Ende des Assembler-Programms durch das Text-Ende des EDIT-Buffers definiert ist. Wird die "END"-Anweisung dennoch benutzt, so wird der Assemblier-Vorgang in dieser Zeile beendet, und es erfolgt eine Rückkehr zum BFZ-EDITOR.

Der "RESTART"-Befehl zur Rückkehr eines Maschinenprogramms in die "USER"-Ebene des MAT 85 läßt sich durch die Assembler-Anweisung

RST 1 ;*** USER ***

direkt erzeugen.

- d) Zur besseren Dokumentation eines Programms sollten entsprechende Kommentare benutzt werden. Die Kommentare werden durch ein vorangestelltes ";" gekennzeichnet, z.B.:

```

                ZK EQU 0FFFF ;ZEITKONSTANTE = FFFF(HEX)
;---          UNTERPROGRAMM: ZEITSCHLEIFE ---
ZEIT:  PUSH B           ;REGISTER RETTEN
        PUSH PSW
        LXI B,ZK       ;BC MIT ZEITKONSTANTE LADEN
Z1:    DCX B           ;ABWAERTS ZAEHLEN
        MOV A,B
        ORA C          ;BC = 0000 ?
        JNZ Z1        ;NEIN: WEITER ABWAERTS ZAEHLEN

        POP PSW       ;REGISTER ZURUECK
        POP B
        RET           ;RUECKSPRUNG

```

Bei der Verwendung von Kommentaren ist zu beachten, daß beim Assemblier-Vorgang vor jede bearbeitete Zeile die entsprechende Adresse mit dem zugehörigen Maschinen-Code geschrieben wird. Damit rückt der Assembler-Code in die Bildschirm-Spalte 17. Wird dadurch die mit einem Kommentar versehene Assembler-Zeile zu lang, so führt dieses zu einem Fehler beim Assemblier-Vorgang mit der entsprechenden Fehlermeldung (siehe auch nächster Abschnitt).

- e) Wurde beim Erstellen des Assembler-Textes gegen die Regeln des MAT 85 - Assemblers verstoßen ("Syntax-Fehler", z.B. MOV AB anstelle von MOV A,B oder doppelt definierte Labels oder ähnliches), so führt dieses beim Assemblieren zu einer Fehlermeldung:

```

--- FEHLER BEIM ASSEMBLIEREN
==> <CR>,<SPACE>

```

Nach der Betätigung von <CR> oder <SPACE> wird die entsprechende Seite aus dem EDIT-Buffer auf dem Bildschirm dargestellt und der Cursor auf den Anfang der fehlerhaften Zeile positioniert. Diese Zeile kann nun korrigiert und der nächste Assemblier-Vorgang gestartet werden. War der Assemblier-Vorgang fehlerfrei, so erfolgt die Meldung:

```

--- ASSEMBLIEREN OK
==> <CR>,<SPACE>

```

Mit der Eingabe von <CR> oder <SPACE> erfolgt dann eine Rückkehr zum BFZ-EDITOR (Warmstart).

- f) Beim Assemblieren mit Bildschirm-Listing wird der Assemblier-Vorgang nach der Ausgabe einer Bildschirmseite unterbrochen. Durch die Eingabe von <CR> oder <SPACE> läßt sich der Assemblier-Vorgang weiter fortführen.
- g) Beim Assemblieren ohne Bildschirm-Listing wird die Bildschirm-Ausgabe unterdrückt. Dadurch lassen sich längere Assembler-Programme wesentlich schneller in den zugehörigen Maschinen-Code umsetzen. Fehler beim Assemblieren führen auch in diesem Fall zu einer Fehlermeldung, die fehlerhafte Zeile wird durch die entsprechende Cursor-Position nach der Rückkehr zum BFZ-EDITOR lokalisiert.
- h) Ein Assemblier-Vorgang mit Drucker-Listing sollte verständlicherweise erst nach der Beseitigung aller Syntax-Fehler erfolgen, da sonst im Fehlerfall der Druckvorgang unterbrochen wird.

```

!!!   Der Ausdruck während des Assemblierens erfolgt immer   !!!
!!!   in Großschreibung.                                     !!!

```

Für die Steuerung des Formularvorschubs (maximale Zeilenanzahl und/oder "Punktzeilen"), sowie für das Ausdrucken von Kopfzeilen ("Punktzeilen" am Anfang des Assembler-Textes) gelten dieselben Regeln wie für das einfache Ausdrucken eines Textes ohne gleichzeitiges Assemblieren. Die verwendeten "Punktzeilen" werden beim Assemblieren unterdrückt, sie dienen nur zur Drucker-Steuerung. Die Einstellung von Zeilenrand und maximaler Zeilenanzahl pro Druck-Seite, sowie die Wahl der Seiten-Numerierung erfolgt mit Hilfe des "Drucker-Menues" (CTRL-D) mit anschließendem EDITOR-Rücksprung.

Beispiel: EDIT-Buffer-Start-Adresse: 6000
 EDIT-Buffer-Stop -Adresse: 6308

```

;-----
;-      BEISPIEL-PROGRAMM:      -
;-      UMWANDLUNG VON KLEIN-   -
;-      IN GROSS-BUCHSTABEN    -
;-----

BUFANF EQU 06000      ; BUFFER-ANFANG
GROSS  EQU 00EE9     ; MAT-85-ROUTINE

      ORG 0E000      ; <-- PROGRAMM-ANFANG

      LXI H,BUFANF   ; HL AUF BUFFER-ANFANG
NEXT:  MOV A,M       ; ZEICHEN HOLEN

      CALL GROSS     ; UMWANDELN IN GROSS-
                   ; BUCHSTABEN

      MOV M,A        ; UND WIEDER SPEICHERN
      INX H          ; ZEIGER ERHOEHEN
      ORA A          ; TESTE AUF 00
      JNZ NEXT       ; NAECHSTES ZEICHEN, WENN
                   ; NICHT 00
      RST 1          ; BUFFER-ENDE: "USER"

      LP

```

Beim Assemblieren des obigen Programm-Beispiels wird der zugehörige Maschinen-Code im Speicherbereich von E000 bis E00D (hex.) abgelegt:

```

0000      ;-----
0000      ; -      BEISPIEL-PROGRAMM:      -
0000      ; -      UMWANDLUNG VON KLEIN-   -
0000      ; -      IN GROSS-BUCHSTABEN    -
0000      ;-----
0000
0000      BUFANF EQU 06000      ; BUFFER-ANFANG
0000      GROSS  EQU 00EE9     ; MAT-85-ROUTINE
0000
0000      ORG 0E000      ; <-- PROGRAMM-ANFANG
E000
E000 21 0060      NEXT:  LXI H,BUFANF   ; HL AUF BUFFER-ANFANG
E003 7E          MOV A,M       ; ZEICHEN HOLEN
E004
E004 CD E90E     CALL GROSS     ; UMWANDELN IN GROSS-
E007          ; BUCHSTABEN
E007
E007 77          MOV M,A        ; UND WIEDER SPEICHERN
E008 23          INX H          ; ZEIGER ERHOEHEN
E009 B7          ORA A          ; TESTE AUF 00
E00A C2 03E0     JNZ NEXT       ; NAECHSTES ZEICHEN, WENN
E00D          ; NICHT 00
E00D CF          RST 1          ; BUFFER-ENDE: "USER"
E00E
E00E          LP
BUFANF 6000
GROSS  0EE9
NEXT   E003
E00E

```

Lösung der Übung 1

 EDITOR-Aufruf, einfache Texteingabe, Cursor-Steuerung
 Untersuchung des EDIT-Buffers, Neustart, Hilfsmenue-Aufruf

- Zu 1.: Nach dem Aufruf des EDITOR-Programms erscheint der EDITOR-Meldetext.
- Zu 2.: Durch Betätigung der ESC-Taste erfolgt eine Rückkehr zum Betriebssystem MAT 85.
- Zu 3.: Durch die Eingabe von "X" läßt sich das Hilfs-Menue aufrufen und durchblättern (Taste <SPACE>). Mit <CR> erfolgt die Rückkehr zum EDITOR-Meldetext.
 Wurde das Hilfs-Menue nicht installiert, so erfolgt eine akustische Fehlermeldung. Der EDITOR-Meldetext bleibt auf dabei auf dem Bildschirm erhalten.
- Zu 4.: Nach dem "Kaltstart" wird als Start-Adresse vorgeschlagen:

START-ADR = 6000

Nach Änderung der Start-Adresse auf 6200 mit anschließendem <CR> oder <SPACE> lautet die Stop-Adresse:

START-ADR = 6000 6200	<CR>
STOP -ADR = 6202	

Begründung: Bei einem Kaltstart werden an den Anfang des EDIT-Buffers die beiden Steuerbytes "0D 0A", sowie die Ende-Markierung "00" eingeschrieben
 ==> Stop-Adresse = 6202

- Zu 5.: Wird nach der Eingabe des angegebenen Textes ein "Warmstart" ausgeführt (CTRL-N mit anschließendem <CR> oder <SPACE>), so liegen die Grenzen des EDIT-Buffers bei:

START-ADR = 6200
STOP -ADR = 622F

Begründung: Durch die Text-Eingabe hat sich das EDIT-Buffer-Ende entsprechend verschoben
 (0D 0A, 26 Zeichen der ersten Zeile, 0D 0A,
 15 Zeichen der zweiten Zeile, 0D 0A, 00 =
 48 (dez.) Zeichen = 30 (hex.) Zeichen
 ==> Stop-Adresse = 622F)

Zu 6.: Nach der Rückkehr zum Monitorprogramm MAT-85 stellt sich der Inhalt des EDIT-Buffers mit Hilfe des PRINT-Kmodus wie folgt dar:

```
6200 0D 0A .D .I .E .S .E .S
6208 . .I .S .T . .E .I .N
6210 .E . .T .E .S .T .Z .E
6218 .I .L .E .. 0D 0A .A .A
6220 .A . . . .B .B .B .
6228 . . .C .C .C 0D 0A 00
6230 .
```

Als Steuerbytes zur Zeilentrennung werden benutzt:

Zeilentrennung:

0D

0A

Als Markierungsbyte am Ende des EDIT-Buffers steht:

Ende-Markierung:

00

Zu 7.: Nach einem erneuten Warmstart mit der vorgeschlagenen Start-Adresse 6200 erscheint der vorher eingegebene Text wieder auf dem Bildschirm.

Nach einem "Neustart" (CTRL-N) mit nachfolgendem "Kaltstart" ab Adresse 6200 ist der alte Text "gelöscht". Die Stop-Adresse des EDIT-Buffers liegt bei 6202.

Auch ein nachfolgender "Warmstart" liefert das gleiche Ergebnis.

Zu 8.: Nach der Rückkehr zum MAT-85 läßt sich der Inhalt des EDIT-Buffers mit Hilfe des PRINT-Modus auf dem Bildschirm darstellen:

```
6200 0D 0A 00 .I .E .S .E .S
```

Die EDIT-Buffer-Endemarkierung 00 ist in den Speicherplatz 6202 eingeschrieben worden.

Mit Hilfe des MEMORY-Modus des MAT-85 läßt sich das 00-Byte unter der Adresse 6202 ändern in 20 (hex.). Wird nun erneut ein Warmstart des EDITOR-Programms ausgeführt, so erscheint der alte Text wieder, mit Ausnahme des ersten Zeichens:

```
IESES IST EINE TESTZEILE.<
AAA   BBB   CCC<
```

Das erste Zeichen wurde durch den Hex-Wert 20 (entspricht ASCII-Leerzeichen) überschrieben. Auf diese Weise läßt sich ein schon vorhandener Text im EDIT-Buffer, der versehentlich durch einen Kaltstart "gelöscht" wurde, wieder restaurieren.

Zu 9.: Nach dem Ausschalten des MFA-Systems und nachfolgendem Wieder-Einschalten wird ein "Power-On-Reset" durchgeführt. Als Start-Adresse für den EDIT-Buffer wird daher vorgeschlagen:

START-ADR = 6000

B. Anpassung des BFZ-EDITORS an andere Terminals

B.1. Terminal-Anpassungs-Tabellen

Sollte bei dem verwendeten MFA-System ein Terminal als Datensicht-, bzw. Eingabestation angeschlossen sein, so ist zusätzlich zur schon vorhandenen Änderung des Betriebsprogramms MAT 85 eine Anpassung des BFZ-EDITOR-Programms an dieses Terminal erforderlich.

Zur einfachen Anpassbarkeit sind innerhalb des EDITOR-Programms Tabellen angelegt, mit deren Hilfe sich die Terminal-spezifischen Parameter verändern lassen. Auf der folgenden Seite sind die Anpassungs-Tabellen für das MFA-Video-Interface dargestellt. Im Anschluß daran finden sich als Beispiel für ein anderes Terminal die Anpassungs-Tabellen für das Terminal "TELEVIDEO".

Um die Terminal-spezifischen Parameter zu verändern, muß das BFZ-EDITOR-Programm in einen freien RAM-Bereich des Arbeitsspeichers kopiert werden. Mit Hilfe des "MEMORY"-Kommandos des Betriebssystems lassen sich die entsprechenden Parameter des verwendeten Terminals in die zugehörigen Tabellenplätze einsetzen. Anschließend kann das EDITOR-Programm mit den veränderten Tabellenwerten in ein neues EPROM programmiert werden.

Die Ermittlung der Terminal-spezifischen Parameter wird im folgenden erläutert.

```
*****
* Tabellen der Steuerbytes für "MFA-Video-Interface" *
*****
```

```
----- TABELLE 1: BILDSCHIRMFORMAT -----
```

ADR.	HEX	DEZ	
5004	3D	61	; SPALTEN (= 64 - 3 dez.)
5005	0F	15	; ZEILEN (= 16 - 1 dez.)
5006	20	} 20000	; ZEITKONSTANTE FUER HOME
5007	4E		

```
----- TABELLE 2: AUSGABE-WERTE -----
```

ADRESSE	HEX	
5008	08	; (OUT)-CURSOR LINKS
5009	09	; " " RECHTS
500A	0A	; " " UNTEN
500B	0B	; " " OBEN
500C	0C	; LOESCHE BILDSCHIRM (HOME)
500D	0D	; LOESCHEN ZEILENREST + CARRIAGE
500E	00	RETURN (00 HIER: OHNE BEDEUTUNG)
500F	1D	; (OUT)-CARRIAGE-RETURN OHNE LOESCHEN DES ZEILENRESTS

```
----- TABELLE 3: EINGABE-WERTE -----
```

ADRESSE	HEX	
5010	08	; (IN)-CURSOR LINKS
5011	09	; " " RECHTS
5012	0A	; " " UNTEN
5013	0B	; " " OBEN
5014	0D	; (IN)-CARRIAGE-RETURN
5015	15	; CTRL-U = SEITE UNTEN
5016	0F	; CTRL-O = SEITE OBEN
5017	05	; CTRL-E = EINFUEGEN EIN/AUS
5018	12	; CTRL-R = RADIERE ZEICHEN
5019	7F	; DEL = DELETE ZEICHEN
501A	1A	; CTRL-Z = ZEILEN-BEFEHLE
501B	14	; CTRL-T = TABULATOR
501C	03	; CTRL-C = ERWEITERTE CURSOR-BEFEHLE
501D	17	; CTRL-W = CURSOR NAECHSTES WORT
501E	0E	; CTRL-N = NEUSTART
501F	18	; CTRL-X = HILFS-MENUE
5020	01	; CTRL-A = ASSEMBLIEREN
5021	04	; CTRL-D = DRUCKER-MENUE
5022	06	; CTRL-F = FIND-MODUS
5023	02	; CTRL-B = BLOCK-BEFEHLE
5024	13	; CTRL-S = SICHERN
5025	00	; MARKIERUNG: TABELLEN-ENDE

```
*****
* Tabellen der Steuerbytes für Terminal "TELEVIDEO" *
*****
```

```
----- TABELLE 1: BILDSCHIRMFORMAT -----
```

```
ADRESSE  HEX  DEZ

5004      4D   77      ; SPALTEN (= 80 - 3 dez.)
5005      17   23      ; ZEILEN  (= 24 - 1 dez.)
5006      00   }
5007      00   } 0      ; ZEITKONSTANTE FUER HOME
```

```
----- TABELLE 2: AUSGABE-WERTE -----
```

```
ADRESSE  HEX

5008      08      ; (OUT)-CURSOR LINKS
5009      0C      ; "      "      RECHTS
500A      0A      ; "      "      UNTEN
500B      0B      ; "      "      OBEN
500C      1A      ; LOESCHE BILDSCHIRM (HOME)

500D      1B   }
500E      74   }  ; LOESCHE ZEILENREST (HIER: ESCAPE-
                    SEQUENCE: "ESC t")

500F      0D      ; (OUT)-CARRIAGE-RETURN OHNE
                    LOESCHEN DES ZEILENRESTS
```

```
----- TABELLE 3: EINGABE-WERTE -----
```

```
ADRESSE  HEX

5010      08      ; (IN)-CURSOR LINKS
5011      0C      ; "      "      RECHTS
5012      16      ; "      "      UNTEN
5013      0B      ; "      "      OBEN
5014      0D      ; (IN)-CARRIAGE-RETURN

5015      15      ; CTRL-U = SEITE UNTEN
5016      0F      ; CTRL-O = SEITE OBEN
5017      05      ; CTRL-E = EINFUEGEN EIN/AUS
5018      12      ; CTRL-R = RADIERE ZEICHEN
5019      7F      ; DEL    = DELETE ZEICHEN
501A      1A      ; CTRL-Z = ZEILEN-BEFEHLE
501B      09      ; TAB    = TABULATOR
501C      03      ; CTRL-C = ERWEITERTE CURSOR-BEFEHLE
501D      17      ; CTRL-W = NAECHSTES WORT
501E      0E      ; CTRL-N = NEUSTART
501F      18      ; CTRL-X = HILFS-MENUE
5020      01      ; CTRL-A = ASSEMBLIEREN
5021      04      ; CTRL-D = DRUCKER-MENUE
5022      06      ; CTRL-F = FIND-MODUS
5023      02      ; CTRL-B = BLOCK-BEFEHLE
5024      13      ; CTRL-S = SICHERN
5025      00      ; MARKIERUNG: TABELLEN-ENDE
```

B.2. Ermittlung der Terminal-spezifischen Parameter:

B.2.1. Tabelle 1: Bildschirmformat

Mit Hilfe dieser Tabelle wird das EDITOR-Programm auf das Bildschirmformat des verwendeten Terminals eingestellt (Bildschirmformat: Spalten- und Zeilenzahl).

Die Berechnung der Spalten- und Zeilenzahl für das BFZ-EDITOR-Programm erfolgt nach folgender Vorschrift:

5004	Spalten	=	max. Bildschirmspalten	-	3
5005	Zeilen	=	max. Bildschirmzeilen	-	1

Für das MFA-Video-Interface ergibt sich somit:

5004	Spalten	=	64	-	3	=	61 (Dez)	=	3D (Hex)
5005	Zeilen	=	16	-	1	=	15 (Dez)	=	0F (HEX)

Für das verwendete Terminal mit einer Bildschirm-Darstellung von

.... Spalten x Zeilen

ergibt sich damit:

5004	Spalten	=	-	3	=	=
5005	Zeilen	=	-	1	=	=

Das MFA-Video-Interface benötigt zur Ausführung des Steuerbefehls "HOME" (Bildschirm löschen, Cursor an linken oberen Bildrand) eine Zeit von ca. 0,21 s. Aus diesem Grund muß nach der Ausgabe des "HOME"-Befehls eine entsprechende Zeitverzögerung gewährleistet werden, bevor das nächste Zeichen ausgegeben wird. Diese Zeitverzögerung, die bei anderen Terminals unter Umständen nicht notwendig ist, läßt sich ebenfalls mit Hilfe der Tabelle 1 ändern.

Die erforderliche Zeitkonstante für die Ausführung von "HOME" muß für das MFA-Video-Interface wie folgt gewählt werden:

ZK-HOME = 20000 (DEZ) = 4E20 (HEX)

Wird für das verwendete Terminal keine Zeitverzögerung benötigt, so kann als Zeitkonstante verwendet werden:

ZK-HOME = 0 (DEZ) = 0000 (HEX)

Für das verwendete Terminal wird als Zeitkonstante benötigt (experimentell ermittelt oder aus Terminal-Handbuch):

5006 (L-Byte) (DEZ)
5007 (H-Byte)	

B.2.2. Tabelle 2: Ausgabewerte

Zur Steuerung der Cursor-Bewegung auf dem Bildschirm werden vom EDITOR-Programm bestimmte Steuer-Bytes zur Datensichtstation ausgegeben. Diese Terminal-spezifischen Ausgabewerte sind in der Tabelle 2 abgelegt.

Als erstes muß ermittelt werden, welche Tastatureingaben für die Steuerung der Cursor-Bewegung, für das Löschen des Zeilenrests, sowie für die "HOME"-Funktion benötigt werden. Dieses kann mit Hilfe des Terminals-Handbuches oder mit dem nachfolgenden Programm 1 geschehen:

```

;---- PROGRAMM 1 ----
                                ORG 0E000
L1:  CALL 0FC80                ;TASTATUR ABFRAGEN
      ANI 7F                   ;BIT 7 WEGMASKIEREN
      CALL 0052                ;ZEICHEN ZUR DATENSICHTSTATION
      JMP L1

```

Als Tastatureingaben können unter Umständen auch "CTRL"-Eingaben oder auch "ESC-Sequenzen" in Frage kommen (z.B. CTRL-L für "HOME" oder "ESC t" (= 1B 74 hex.) für "Löschen des Zeilenrests").

Bei der Ermittlung der erforderlichen Tastatureingabe für die Cursor-Bewegung nach unten ist zu beachten, daß diese Eingabe zu einem "Scrollen" des Bildschirminhalts nach oben führt, wenn sich der Cursor in der untersten Bildschirmzeile befindet.

Tragen Sie in nachfolgende Liste ein, welche Tastatureingabe für die entsprechende Steuerfunktion erforderlich ist:

	Funktion	MFA- Video-Interface		Terminal	
		Tastatur- Eingabe	Steuer- byte	Tastatur- Eingabe	Steuer- byte
5008	Cursor links	:	Pfeil links	08
5009	Cursor rechts	:	Pfeil rechts	09
500A	Cursor unten (mit "Scrollen")	:	Pfeil unten	0A
500B	Cursor oben	:	Pfeil oben	0B
500C	HOME	:	CTRL-L	0C
500D	Löschen des Zeilen- rests, evtl. mit	}	<CR>	0D
500E	Carriage-Return		NULL	00
500F	Carriage-Return ohne Löschen Zeilenrest	:	CTRL-]	1D

Die zu den Tastatureingaben zugehörigen Steuer-Bytes lassen sich anschließend mit Hilfe des folgenden Programms 2 ermitteln und in obige Liste eintragen.

```

;---- PROGRAMM 2 ----
                ORG 0E100

L2:             CALL 0FC80           ; TASTATUR ABFRAGEN
                ANI 7F              ; BIT 7 WEGMASKIEREN
                CALL 0058           ; HEX-WERT AUSGEBEN
                CALL 0070           ; CR-LF AUSGEBEN
                JMP L2
    
```

B.2.3. Tabelle 3: Eingabewerte

Die von der Tastatur bei der Betätigung einer Taste ausgegebenen Steuerbytes (Eingabewerte) sind ebenfalls Terminal-spezifisch. Mit Hilfe des obigen Programms 2 läßt sich ermitteln, welche Steuerbytes bei einer Tastenbetätigung von der Tastatur erzeugt und zum MFA-Computer übertragen werden.

Ermitteln Sie die zu den Tastatureingaben zugehörigen Steuerbytes und tragen Sie diese in nachfolgende Liste ein (dabei darf kein Steuerbyte doppelt vorhanden sein).

Adr.	EDITOR-Funktion	MFA-Video-Interface		Terminal	
		Tastatur-Eingabe	Steuerbyte	Tastatur-Eingabe	Steuerbyte
5010	Cursor links	Pfeil links	08
5011	Cursor rechts	Pfeil rechts	09
5012	Cursor unten	Pfeil unten	0A
5013	Cursor oben	Pfeil oben	0B
5014	Cursor an Zeilenanfang	<CR>	0D
5015	Seite unten	CTRL-U	15
5016	Seite oben	CTRL-O	0F
5017	Einfügen EIN/AUS	CTRL-E	05
5018	Radiere Zeichen	CTRL-R	12
5019	Delete Zeichen	DEL	7F
501A	Zeilen-Befehle	CTRL-Z	1A
501B	Tabulator	CTRL-T	14
501C	Erw. Cursorbew.	CTRL-C	03
501D	Nächstes Wort	CTRL-W	17
501E	Neustart	CTRL-N	0E
501F	Hilfs-Menue	CTRL-X	18
5020	Assemblieren	CTRL-A	01
5021	Drucker-Menue	CTRL-D	04
5022	Find-Modus	CTRL-F	06
5023	Block-Befehle	CTRL-B	02
5024	Sichern	CTRL-S	13

C. Aufbau des Hilfs-Menues

Um spätere Änderungen des Hilfs-Menues durch den Anwender zu ermöglichen, wird im folgenden der Aufbau des Hilfs-Menues erläutert.

Das Hilfs-Menue kann an jede beliebige freie Stelle des Arbeitsspeichers geladen werden. Ist das verwendete MFA-System nicht mit dem "BFZ-MINI-DDS" bestückt, so läßt sich das Hilfs-Menue als EPROM-Version ab der Speicherplatzadresse 4000 (HEX) installieren, und der BFZ-EDITOR kann aus der Kommando-Ebene des MAT 85 heraus mit Hilfe der Tastatur-Eingabe "E" (= EDITOR) gestartet werden.

Dazu ist in den ersten drei Speicherplätzen des Hilfs-Menues ein Sprungbefehl zur Anfangsadresse des EDITOR-Programms (5000 HEX) abgelegt:

4000	C3	
4001	00	JMP 5000
4002	50	

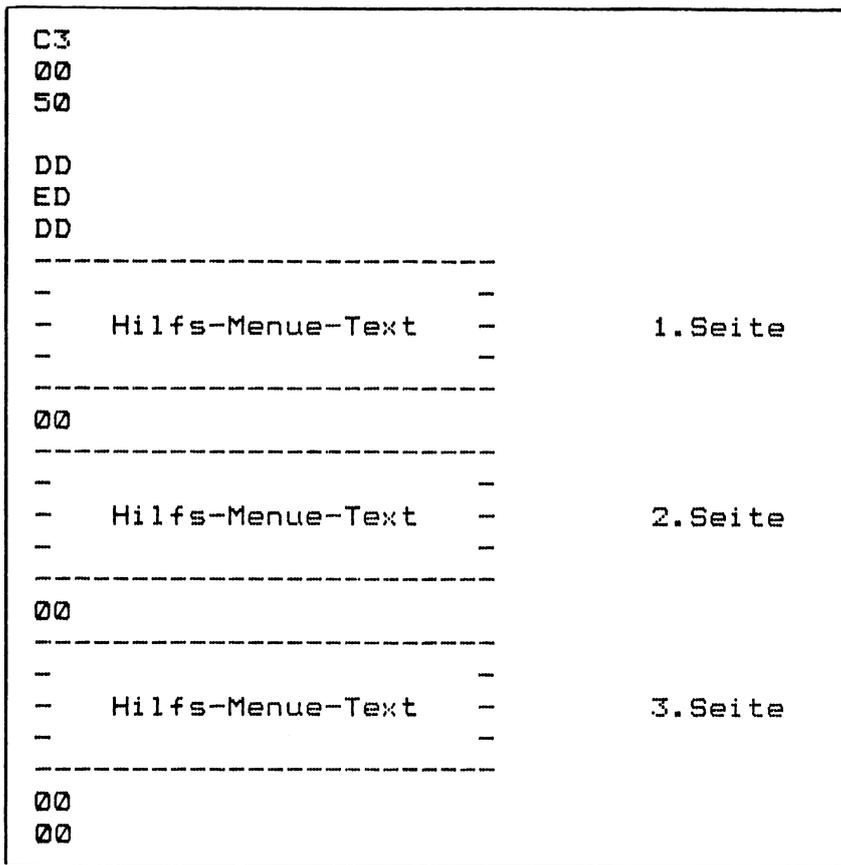
(Wird innerhalb der Kommando-Ebene des Betriebssystems MAT 85 eine Tastatureingabe vorgenommen, die nicht in der Kommando-Liste des MAT 85 enthalten ist, so überprüft das Betriebsprogramm, ob in dem Speicherplatz 4000 (HEX) ein Sprungbefehl ("C3") enthalten ist. Ist dies der Fall, so wird der Sprungbefehl ausgeführt, das EDITOR-Programm also gestartet. Dieses wird nur dann weiter abgearbeitet, wenn die Tastatureingabe ein "E" war. Ansonsten erfolgt ein Rücksprung zum Betriebssystem.)

In den nachfolgenden drei Speicherplätzen des Hilfs-Menues steht die sogenannte "Hilfs-Menue-Kennung":

4003	DD
4004	ED
4005	DD

Aufgrund dieser Kennzeichnung findet das EDITOR-Programm das Hilfs-Menue an jeder beliebigen Stelle des Arbeitsspeichers. Bei einem Aufruf des Hilfs-Menues vom EDITOR aus wird nach dem Auffinden der Kenn-Bytes der ab der nachfolgenden Adresse gespeicherte Text auf dem Bildschirm dargestellt.

Das Hilfs-Menue hat demnach folgenden Aufbau:

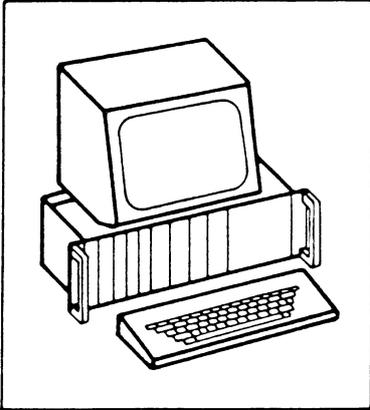


Findet das EDITOR-Programm bei der Darstellung einer Hilfs-Menue-Seite ein "00"-Byte, so wird die Bildschirmausgabe unterbrochen und die Tastatur abgefragt. Durch die Eingabe von <SPACE> läßt sich die nachfolgende Seite darstellen, bei der Eingabe von <CR> erfolgt eine Rückkehr zum EDITOR. Das Ende des Hilfs-Menues ist mit zwei "00"-Bytes gekennzeichnet. Hier erfolgt bei der Eingabe von <SPACE> eine erneute Darstellung des Hilfsmenus ab der ersten Hilfs-Menue-Seite, die Eingabe von <CR> führt auch hier zu einem Rücksprung in den EDIT-Modus.

Die Veränderung der einzelnen Hilfs-Menue-Seiten kann mit Hilfe des BFZ-EDITORS erfolgen. Die einzelnen Seiten des veränderten Hilfs-Menues müssen anschließend gemäß dem oben beschriebenen Aufbau zusammengefügt werden.

Es bleibt dem Anwender überlassen, unter Umständen auch andere Informationen innerhalb des Hilfs-Menues zu speichern, die dann bei Bedarf jederzeit vom Bediener abrufbar sind.

FACHPRAKTISCHE ÜBUNG MIKROCOMPUTER-TECHNIK



64-KByte-RAM

BFZ/MFA 3.3.



Inhaltsverzeichnis

Inhalt	Seite
Hinweise	1
Blockschaltbild	2
Funktionsbeschreibung	3
Stromlaufplan	7
Bestückung Leiterplatte	8
Inbetriebnahme	10
Bereitstellungsliste	13
Bohrplan Leiterplatte	15
Layout Bestückungsseite	16
Beschriftung Frontplatte	17
Zusammenbau Baugruppe	18

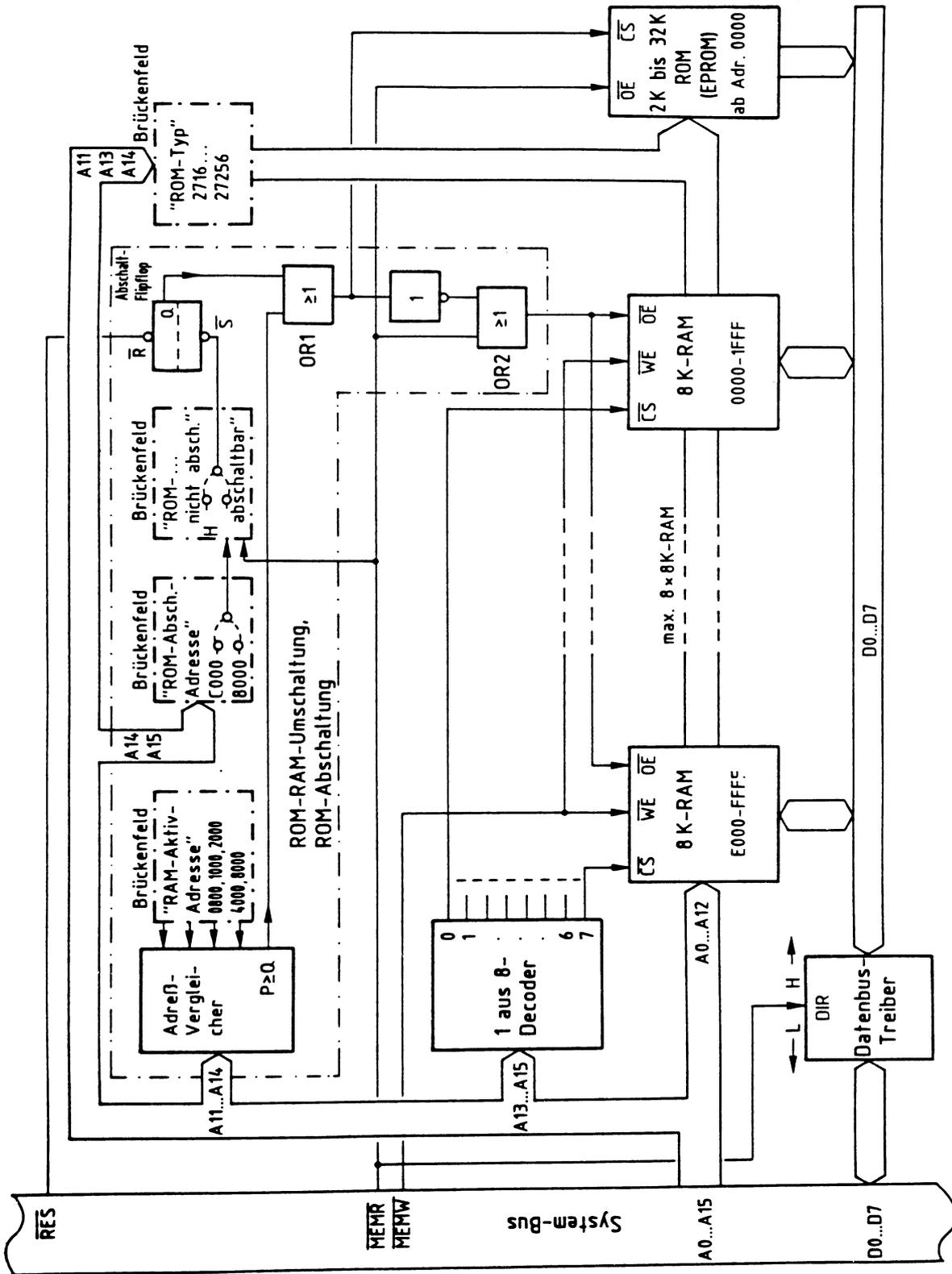
Hinweise

Die 64-K-RAM-Baugruppe ist für den Betrieb des BFZ/MFA-Mikrocomputers unter dem Betriebssystem MAT32-K/S oder MAT32-K/Terminal vorgesehen.

Die Baugruppe enthält vier 8-KByte-RAM-Bausteine.

Das EPROM MAT32-K/S bzw. MAT32-K/Terminal (siehe Ausbaustufe MT bzw. PC) mit einer Speicherkapazität von 32-KByte wird auf dem Steckplatz für IC Ø eingesetzt.

Blockschaltbild



Funktionsbeschreibung

Die Funktionsbeschreibung bezieht sich auf das Blockschaltbild der Baugruppe. Schaltungseinzelheiten sind dem Stromlaufplan zu entnehmen.

Die 64-KByte-RAM-Baugruppe kann wahlweise in der Betriebsart "ROM/RAM-Betrieb, ROM nicht abschaltbar" oder "ROM/RAM-Betrieb mit abschaltbarem ROM" eingesetzt werden. Die Einrichtung der Baugruppe für die gewünschte Betriebsart erfolgt durch Lötbrücken auf mehreren Brückenfeldern. Alle hierzu erforderlichen Angaben (technische Merkmale, typische Speicheraufteilungen) befinden sich am Ende der Funktionsbeschreibung.

Datenbus-Treiber, 1 aus 8-Decoder und Speicher

Der Daten-Bus der Baugruppe ist über einen Datenbus-Treiber mit dem System-Bus verbunden. Die Arbeitsrichtung des Datenbus-Treibers wird mit dem Signal $\overline{\text{MEMR}}$ gesteuert.

Mit den Adreßleitungen A0 bis A12 werden die einzelnen Speicherzeilen der RAM-Bausteine und des EPROMs adressiert. Je nach verwendetem EPROM (2716...27256) werden die Adreßleitungen A11, A13 und A14 im Brückenfeld "ROM-Typ" an die unterschiedlichen Anschlußbelegungen der EPROMs angepaßt.

Die Adreßleitungen A13 bis A15 führen auf einen 1 aus 8-Decoder, der jeweils einen der 8-KByte-RAM-Bausteine über dessen $\overline{\text{CS}}$ -Anschluß freigibt.

Die $\overline{\text{OE}}$ -Anschlüsse aller RAM-Bausteine sind zusammengeführt. Daten des RAM-Speichers können nur dann gelesen werden, wenn die $\overline{\text{OE}}$ -Anschlüsse L-Pegel erhalten. Die Schreib-Lese-Umschaltung der RAM-Bausteine erfolgt über $\overline{\text{MEMW}}$.

ROM-RAM-Umschaltung

Zur ROM-RAM-Umschaltung gehören der Adreßvergleich mit dem Brückenfeld "RAM-Aktiv-Adresse", die ODER-Verknüpfung "OR1", der Inverter und die ODER-Verknüpfung "OR2".

Beim Einschalten des Mikrocomputers (oder bei Betätigung der RESET-Taste) wird das Abschalt-Flipflop über das Signal $\overline{\text{RES}}$ zurückgesetzt, so daß der Ausgang Q an die ODER-Verknüpfung OR1 L-Pegel abgibt.

Funktionsbeschreibung

Der zweite Eingang von OR1 erhält nach dem Einschalten vom Ausgang $P \geq Q$ des Adreßvergleichers ebenfalls L-Pegel, da die vom Prozessor ausgegebene Adresse 0000 stets kleiner als die mit dem Brückenfeld "RAM-Aktiv-Adresse" einstellbare Vergleichsadresse ist. Die ODER-Verknüpfung OR1 gibt daher zunächst L-Pegel ab. Hiermit wird der \overline{CS} -Anschluß des EPROMs angesteuert und das EPROM aktiviert.

Bei aktivem EPROM muß der RAM-Speicher abgeschaltet sein. Die RAM-Abschaltung erfolgt durch H-Pegel an den \overline{OE} -Anschlüssen der RAM-Bausteine. Dieser H-Pegel stammt von der ODER-Verknüpfung OR2 und wird durch Invertierung des Ausgangssignals von OR1 (\overline{CS} -Signal für das EPROM) gebildet.

Durch das beschriebene Verfahren wird nach dem Einschalten bzw. RESET stets zunächst das EPROM aktiviert und der RAM-Speicher abgeschaltet.

Mit dem Brückenfeld "RAM-Aktiv-Adresse" wird eingestellt, ab welcher Speicheradresse statt des EPROMs der RAM-Speicher aktiv sein soll. Einstellbar sind die Adreßwerte 0800, 1000, 2000, 4000 und 8000. Der einzustellende Wert richtet sich nach dem EPROM-Typ.

Sobald der Prozessor eine Adresse ausgibt, die gleich oder größer der eingestellten RAM-Aktiv-Adresse ist, führt der Adreßvergleicher am Ausgang $P \geq Q$ H-Pegel und schaltet über OR1 das EPROM ab.

Zum Lesen von Daten aus dem Speicher steuert der Prozessor die \overline{MEMR} -Leitung auf L-Pegel. Da das EPROM z.Zt. nicht aktiviert ist, führen jetzt beide Eingänge von OR2 L-Pegel, wodurch die \overline{OE} -Anschlüsse der RAM-Bausteine L-Pegel erhalten und der RAM-Speicher zum Lesen von Daten freigegeben ist.

Zum Einschreiben von Daten in den RAM-Speicher ist L-Pegel an den \overline{OE} -Anschlüssen nicht erforderlich.

Arbeitet der Prozessor wieder unterhalb der RAM-Aktiv-Adresse, nimmt der Ausgang $P \geq Q$ des Adreßvergleichers wieder L-Pegel an. Hierdurch wird wieder vom RAM auf das EPROM umgeschaltet. Das Abschalt-Flipflop ist an dieser Umschaltung nicht beteiligt.

Funktionsbeschreibung

ROM-Abschaltung

Beim Einschalten des Mikrocomputers ist zunächst immer das EPROM aktiviert. Für bestimmte Anwendungsfälle der Speicherbaugruppe wie z.B. das Arbeiten unter dem Betriebssystem "CP/M" muß der Speicher bereits ab der Adresse 0000 mit RAM beginnen. Da "CP/M" nicht in EPROMs gespeichert wird, sondern auf einer Diskette, muß das EPROM ein Programm enthalten, mit dessen Hilfe "CP/M" von der Diskette in den RAM-Speicher geladen wird. Nach diesem als "BOOTEN" bezeichneten Ladevorgang wird das EPROM nicht mehr benötigt und muß abgeschaltet werden.

Die Abschaltung des EPROMs erfolgt durch das Abschalt-Flipflop in Verbindung mit den Brückenfeldern "ROM nicht abschaltbar/abschaltbar" und "ROM-Abschalt-Adresse".

Bedingungen für das Abschalten des EPROMs:

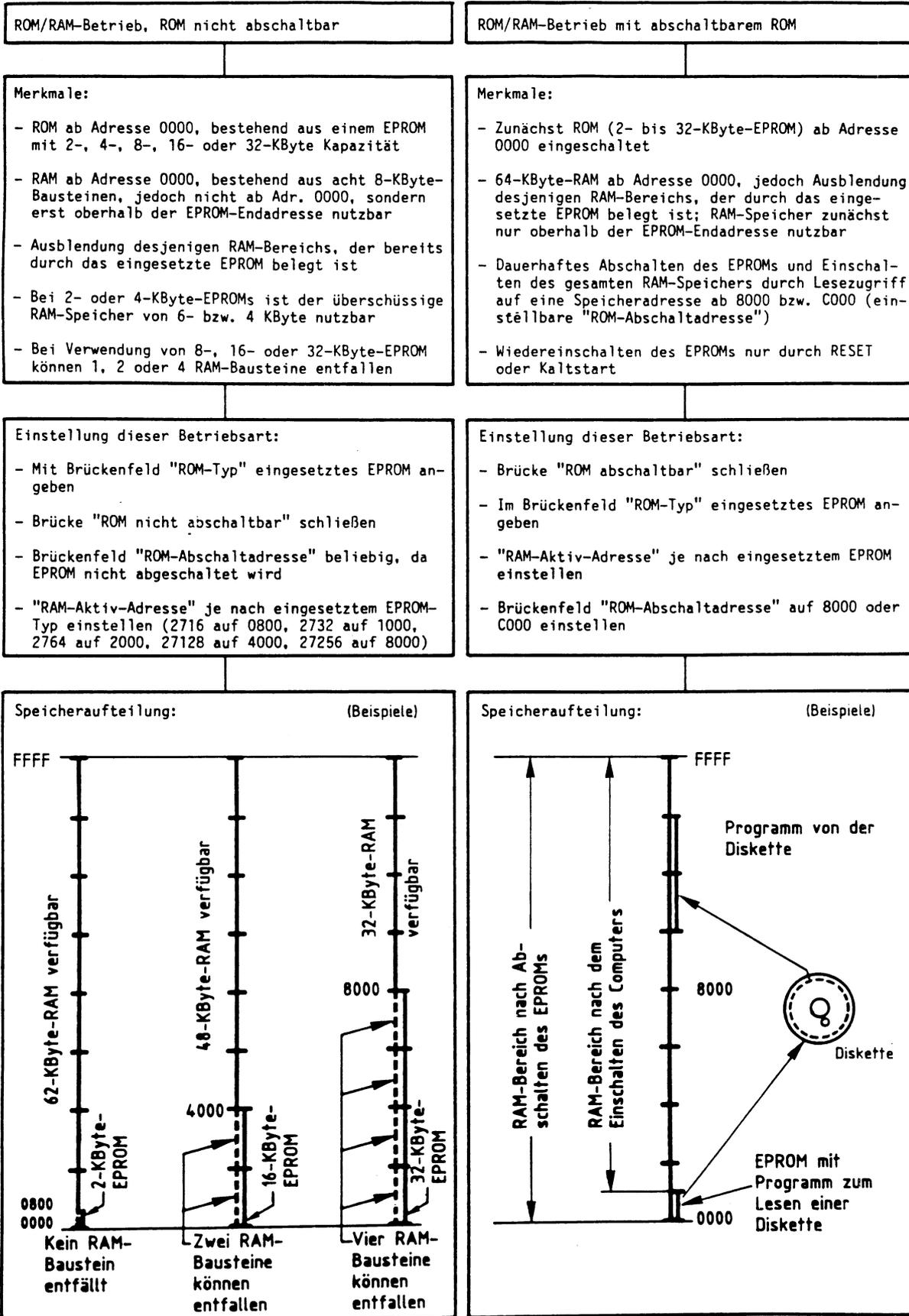
- Brückenfeld "ROM nicht abschaltbar/abschaltbar" in Stellung "abschaltbar"
- Speicher-Lesezugriff des Prozessors auf die mit dem Brückenfeld "ROM-Abschalt-Adresse" eingestellte oder eine höhere Adresse

Nur wenn beide Bedingungen gemeinsam erfüllt sind, wird das Abschalt-Flipflop gesetzt und hierdurch das EPROM dauerhaft abgeschaltet. Anschließend kann auf das EPROM nicht mehr zugegriffen werden. Dieser Zustand bleibt bis zum nächsten RESET erhalten.

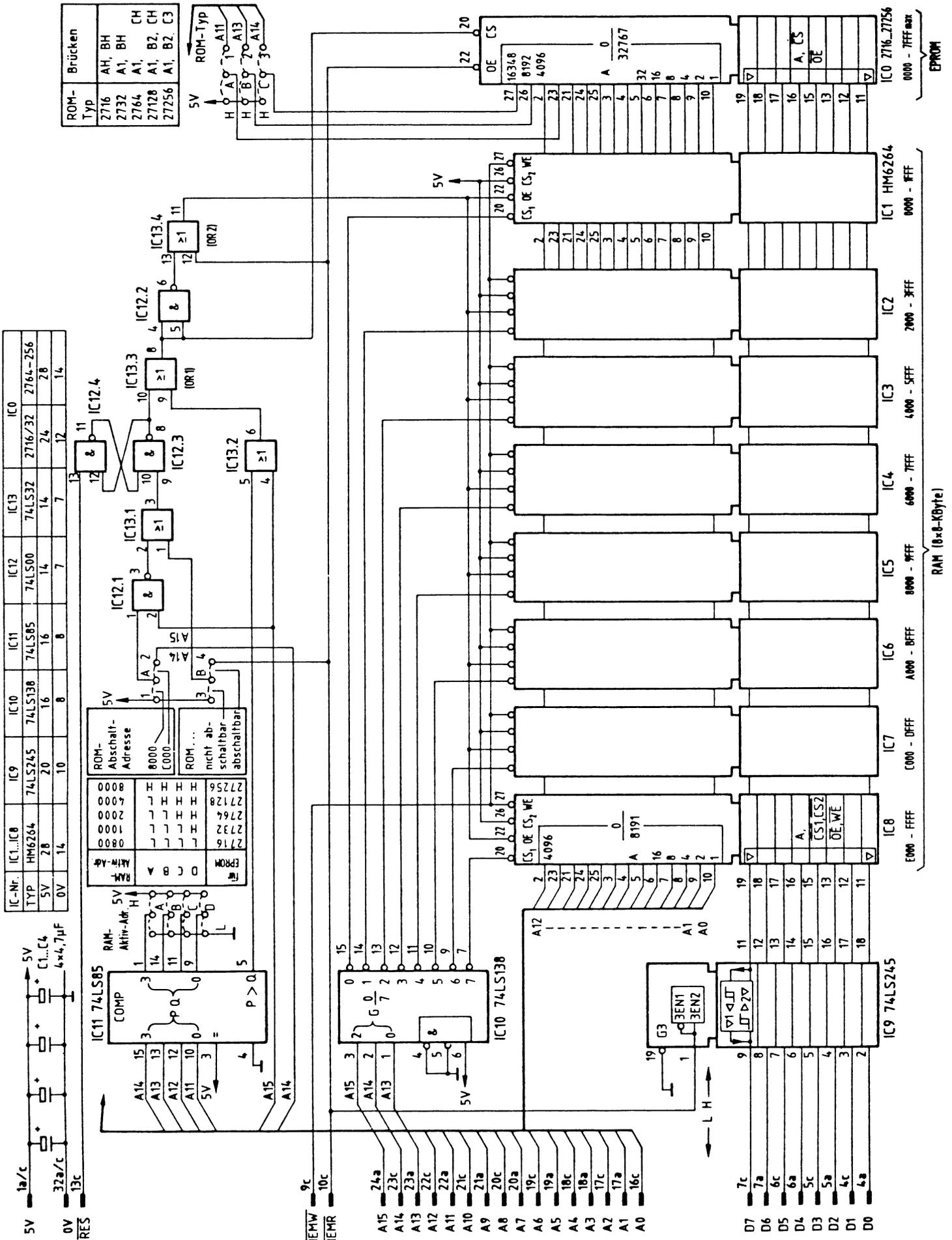
Zusammenfassung

Auf der nächsten Seite sind die Betriebsarten der 64-KByte-RAM-Baugruppe zusammengefaßt dargestellt und Beispiele für einige mögliche Speicheraufteilungen angegeben.

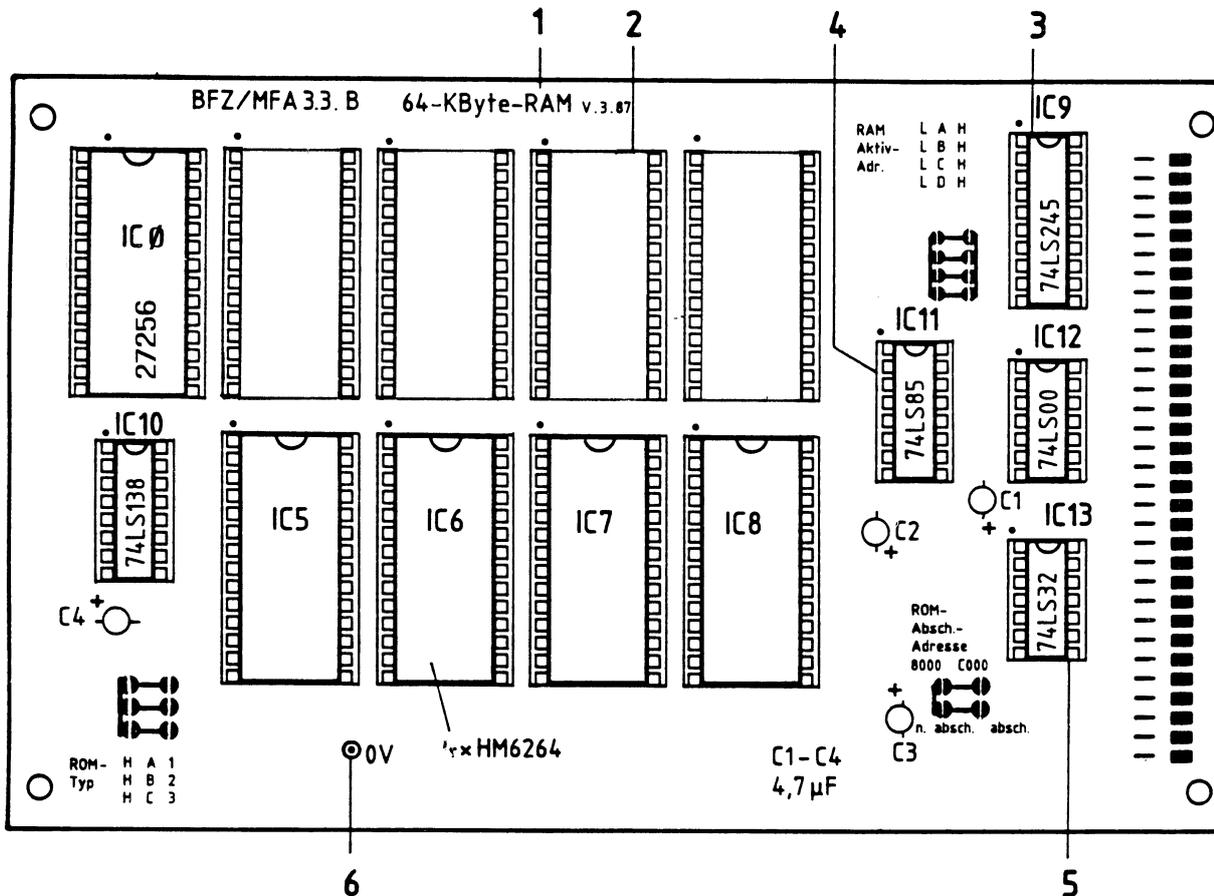
Funktionsbeschreibung



Stromlaufplan



Bestückung Leiterplatte



Stückliste Leiterplatte BFZ/MFA 3.3.

Pos.	Stckz.	Benennung/Daten	Bemerkung
1	1	Leiterplatte BFZ/MFA 3.3.	
2	9	IC-Fassung 28polig	
3	1	IC-Fassung 20polig	
4	2	IC-Fassung 16polig	
5	2	IC-Fassung 14polig	
6	1	Lötstift/-nagel für 1,3 mm Loch-Ø	Meßstift für Masse (0 V)

Bestückung Leiterplatte

Bauteilliste Leiterplatte BFZ/MFA 3.3.

Kennz.	Benennung/Daten	Bemerkung
C1...C4	Tantal-Elko 4,7 μ F/35 V	
IC5...IC8	RAM-Baustein 8K x 8 Bit	
IC9	Acht Bus-Transceiver 74 LS 245	Tri-State
IC10	3-Bit-Binärdekoder 74 LS 138	
IC11	4-Bit-Vergleicher 74 LS 85	
IC12	Vier NAND-Gatter 74 LS 00	
IC13	Vier ODER-Gatter 74 LS 32	

Inbetriebnahme

1. Sichtkontrolle

Die Sichtkontrolle wird anhand von Stromlauf- und Bestckungsplan sowie Stück- und Bauteilliste durchgeführt.

- Sind alle Lötstellen ordnungsgemäß verlötet?
- Sind die Kondensatoren richtig gepolt?
- Sind alle ICs richtig eingesteckt?
- Sind alle Schrauben festgezogen?

2. Prüfen der Funktion

Folgende Lötbrücken müssen geschlossen sein: Im Brückenfeld...

- "RAM-Aktiv-Adr.": A - H, B - H, C - H, D - H
- "ROM-Absch.-Adr.": C000
- "ROM n. absch. - absch.": n. absch.
- "ROM-Typ": A - H, B - H, C - H

2.1. Statische Überprüfung der 64-KByte-RAM-Baugruppe

Die statische Überprüfung erfolgt mit Hilfe eines TTL-Testers oder Vielfach-Meßinstrumentes. Zur Erzeugung und Anzeige der erforderlichen Adreß-, Daten- und Steuersignale dienen Bus-Signalgeber und Bus-Signalanzeige.

- Bestückung des Baugruppenträgers:
- Bus-Signalgeber
 - Bus-Signalanzeige
 - 64-KByte-RAM (über Adapterkarte)

Inbetriebnahme

2.1.1. \overline{CS} der RAMs

- Bus-Signalgeber-Schalter "ON/OFF" in Stellung "ON"
- Adreßeinstellung lt. Tabelle und Überprüfung der \overline{CS} -Signale jeweils an Pin 20 von IC1 bis IC8

ADDRESS (eingest.)	Pegel an den \overline{CS} -Anschlüssen (Pin 20) von ...							
	IC1	IC2	IC3	IC4	IC5	IC6	IC7	IC8
0000	L	H	H	H	H	H	H	H
2000	H	L	H	H	H	H	H	H
4000	H	H	L	H	H	H	H	H
6000	H	H	H	L	H	H	H	H
8000	H	H	H	H	L	H	H	H
A000	H	H	H	H	H	L	H	H
C000	H	H	H	H	H	H	L	H
E000	H	H	H	H	H	H	H	L

2.1.2. Datenflußrichtung, \overline{OE} des EPROMs, \overline{WE} der RAMs

- Bei betätigter MEMR-Taste muß an Pin 1 von IC9 und an Pin 22 von IC0 L-Pegel auftreten.
- Bei betätigter MEMW-Taste muß jeweils an Pin 27 von IC1 bis IC8 L-Pegel auftreten.

Inbetriebnahme

2.1.3. EPROM-Abschalt-Logik

- CPU-Baugruppe zusätzlich im Baugruppenträger
- Bus-Signalgeber-Schalter "ON/OFF" in Stellung "ON"
- Einmal Taste "RESET" betätigt
- Messung der Pegel lt. Tabelle (Reihenfolge einhalten!)

ADRESS (eingest.)	Taste MEMR	IC 12.3 Pin 8	IC 13.2 Pin 6	IC 0 Pin 20	IC 1 Pin 22	IC 5 Pin 22	Bemerkung
0000	n.b.	L	L	L	H	H	EPROM aktiv
0000	bet.	L	L	L	H	H	
7FFF	n.b.	L	H	H	H	H	EPROM n.abgesch.
7FFF	bet.	L	H	H	L	L	
8000	n.b.	L	H	H	H	H	EPROM abgesch.
8000	bet.	H	H	H	L	L	
0000	n.b.	H	L	H	H	H	RAM aktiv
0000	bet.	H	L	H	L	L	

n.b. = nicht betätigt

bet. = betätigt

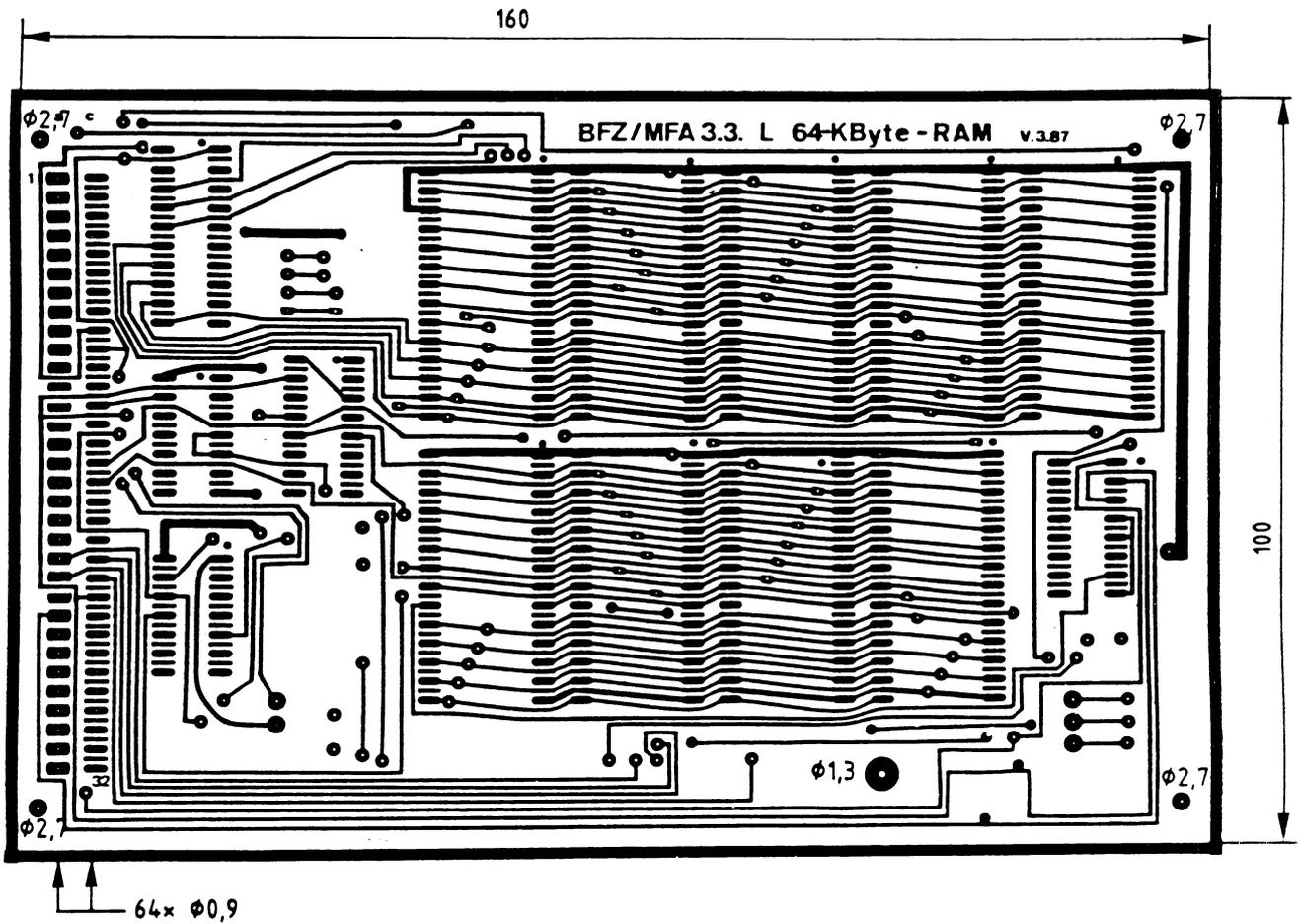
Bereitstellungsliste

Stckz.	Benennung/Daten	Bemerkung
1	Leiterplatte, ca. 110x170 mm Mat.: Epoxid-Glashartgewebe (Hgw 2372)	doppelseitig Cu-kaschiert (35 µm) u. mit Fotolack beschichtet
je 1	Filmvorlage BFZ/MFA 3.3.L u. 3.3.B zum Belichten der Leiterplatte	je nach Ätzverfahren Pos.- oder Neg.-Film
1	Frontplatte, Teilung L-C 05 Alu, 2 mm dick, Breite: 25,1 mm	z.B. Intermas Nr. 409-017 665
1	Griff komplett mit Abdeckung T03	z.B. Intermas Nr. 409-017 927
1	Frontverbinder 1,6 FEE	z.B. Intermas Nr. 409-024 830
1	Messerleiste 64polig, DIN 41612	z.B. Erni STV-P-364 a/c Nr. 9722.333.401
1	Zylinderschraube M2,5x8 DIN 84	
2	Zylinderschraube M2,5x10 DIN 84	
3	Zylinderschraube M2,5x12 DIN 84	
2	Zylinderschraube mit Schaft BM2,5x10/5 DIN 84	
2	Schraubensicherung, Kunststoff	z.B. Intermas Nr. 409-026 748
5	Federscheibe A2,7 DIN 137	
1	Federring B2,5 DIN 127	
4	Sechskantmutter M2,5 DIN 439	
4	Tantal-Elko 4,7 µF/35 V	Tropfenform
1	IC 74 LS 00, Vier NAND-Gatter	
1	IC 74 LS 32, Vier ODER-Gatter	
1	IC 74 LS 85, 4-Bit-Vergleicher	
1	IC 74 LS 138, 3-Bit-Binärdekoder	
1	IC 74 LS 245, Acht Bus-Transceiver	Tri-State

Bereitstellungsliste

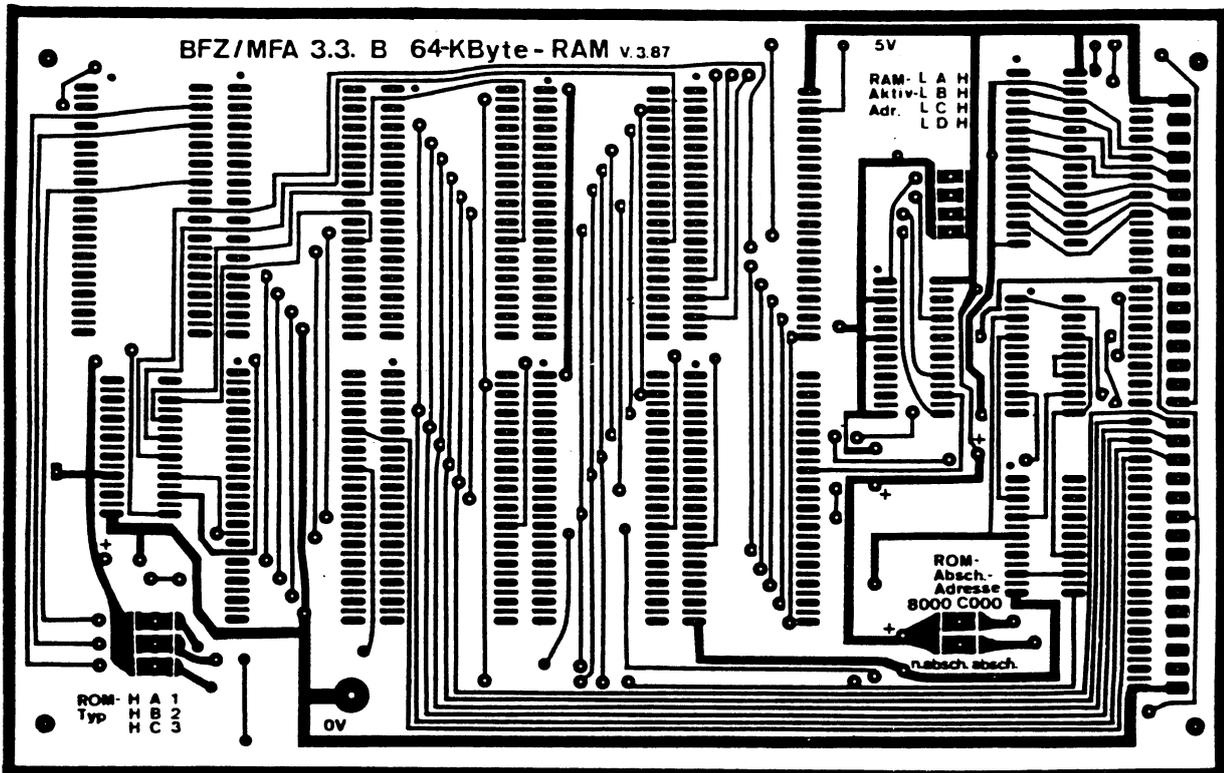
Stckz.	Benennung/Daten	Bemerkung
4	RAM-Baustein 8K x 8 Bit, statisch, max. 300 ns, pinkompatibel mit 6264	z.B. HM 6264 MB 8464-15L
9	IC-Fassung 28polig DIL	
1	IC-Fassung 20polig DIL	
2	IC-Fassung 16polig DIL	
2	IC-Fassung 14polig DIL	
1	Lötstift/Lötnagel für 1,3 mm Loch-Ø	Meßstift für Masse (0 V)

Bohrplan Leiterplatte

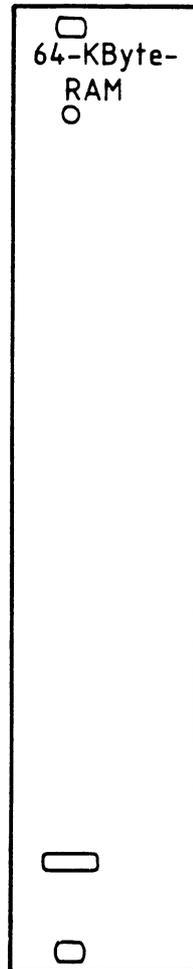


Alle nicht bemaßten Bohrungen $\phi 0,8$ mm
Benötigte Bohrer: 0,8 - 0,9 - 1,3 - 2,7 mm

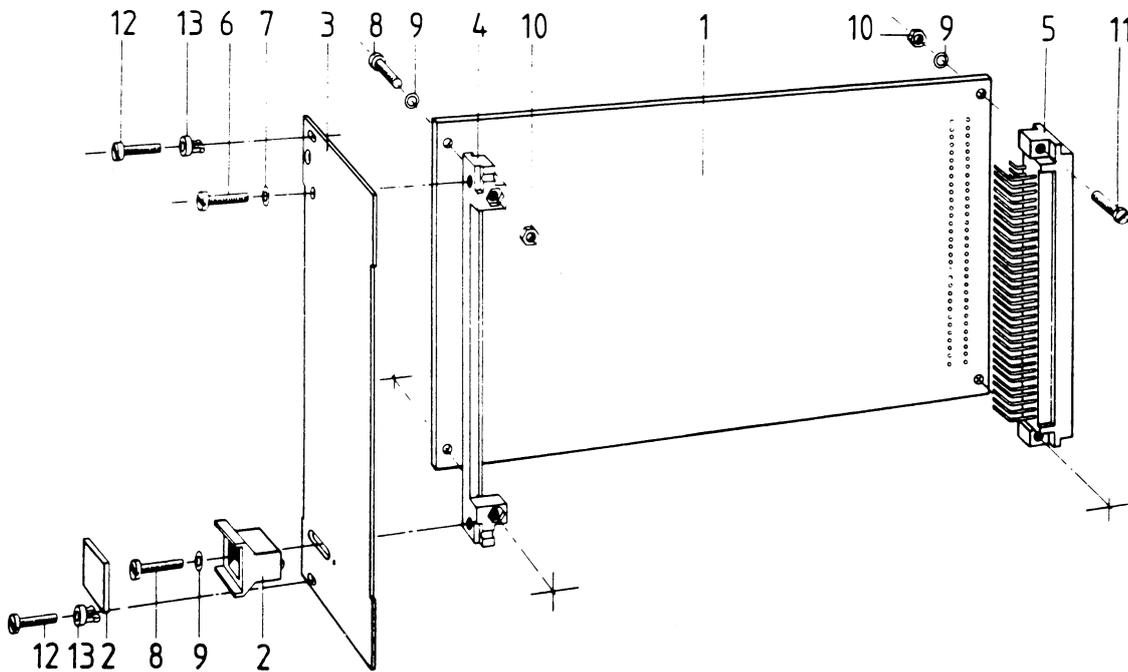
Layout Bestückungsseite



Beschriftung Frontplatte



Zusammenbau Baugruppe



Stückliste für den Zusammenbau

Pos.	Stckz.	Benennung/Daten	Bemerkung
1	1	Leiterplatte BFZ/MFA 3.3.	komplett bestückt
2	1	Griff komplett	
3	1	Frontplatte	
4	1	Frontverbinder	
5	1	Messerleiste 64polig, DIN 41612	
6	1	Zylinderschraube M2,5x8 DIN 84	
7	1	Federring B2,5 DIN 127	
8	3	Zylinderschraube M2,5x12 DIN 84	
9	5	Federscheibe A2,7 DIN 137	
10	4	Sechskantmutter M2,5 DIN 439	
11	2	Zylinderschraube M2,5x10 DIN 84	
12	2	Zylinderschraube mit Schaft, BM2,5x10/5 DIN 84	
13	2	Schraubensicherung, Kunststoff	

MFA-MEDIENSYSTEM

Mikrocomputer- Technik

Fachpraktische Übungen
Band 4

4



VGS Bfz