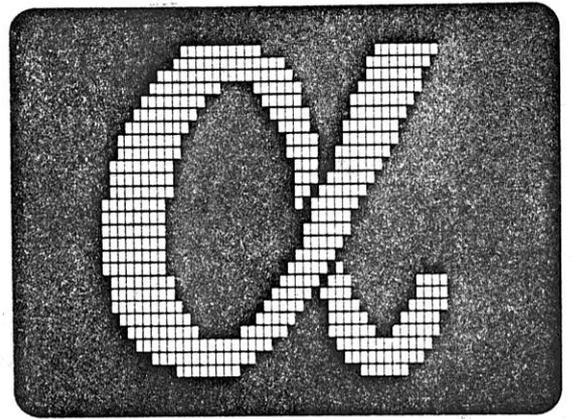


Computer Systeme



**alphaTronic**

Systembeschreibung

	Seite
<b>ALLGEMEINES</b>	
Einleitung .....	1
Zu diesem Manual .....	2
Systembeschreibung .....	3
Expansionsmöglichkeiten .....	4
Softwarewerkzeuge .....	5
Literaturhinweis .....	7
Varianten .....	8
<b>INBETRIEB- NAHME</b>	
Anschlüsse .....	14
Bildschirmanschluß .....	15
Druckeranschluß .....	16
Reset .....	17
Initialisieren .....	18
Formatieren .....	19
Duplizieren .....	20
<b>ZUM AUFWÄRMEN</b>	
Tastenfunktionen .....	21
Laden eines Programmes .....	22
Kommandos / Programmbefehle .....	24
Hardware-Grundlagen .....	25
Software-Grundlagen .....	26
<b>MOS-MONTIOR</b>	
Allgemeines .....	27
Arbeitsweise .....	28
Speicherbelegung .....	29
<b>MOS-Kommandos</b>	
Syntax .....	30
ASSIGN .....	31
E/A-Struktur .....	32
PRESET DEVICE .....	33
OUTPUT .....	35
ENDFILE .....	36
INPUT .....	36
GO .....	37
SUBSTITUTE .....	37
LIST COMMAND OUTPUT .....	38
MOVE MEMORY .....	38
DISPLAY MEMORY .....	38
FILL MEMORY .....	39
COMPARE MEMORY .....	39
TOP OF MEMORY .....	39
EXTENDED MOS .....	39
BATCH MODE .....	40
USER PROGRAM .....	40
I O - BYTE .....	41



alphaTronic

# INHALT

	Seite
<b>BASIC COMPILER</b>	
Allgemeines .....	42
Einführung .....	42
Laden des Compilers .....	43
Speicherbelegung und Arbeitsweise .....	44
Zeichen / Sonderzeichen .....	45
<b>Basic-Kommandos</b>	
Allgemeines .....	46
Syntax .....	46
MOS-Kommandos .....	46
LOAD/SAVE .....	47
NEW / OLD .....	47
LIST .....	48
DELETE .....	48
DUMP .....	48
RENUMBER .....	48
BEGIN .....	49
COMPILE .....	49
RUN .....	50
START .....	50
HEX .....	50
LINK .....	51
CALCULATE / INF / Q / X .....	51
<b>Basic-Befehle</b>	
Syntax .....	52
Sprachumfang .....	53
Zahlen, Zeichen und Variablen .....	54
Gleitkomma-Zahlen .....	54
Festkomma-Zahlen .....	55
String's .....	55
Zahlenvariablen .....	56
Stringvariablen .....	57
DIM .....	58
CHAR .....	58
INTEGER .....	58
LET .....	58
GOTO / GOSUB .....	59
RETURN .....	59
FOR ... NEXT .....	59
IF ... THEN .....	60
REM .....	60
END .....	60
Arithmetische Operatoren .....	61
Vergleichs-Operatoren .....	61
Logische Operatoren .....	61
Rangfolge .....	62

	Seite
<b>Basic-Befehle</b>	
Mathematische Funktionen .....	62
Sonder-Funktionen .....	62
PRINT .....	63
PRINT DEV .....	64
PRINT TAB .....	65
INPUT .....	65
CIS .....	65
CALL-Routinen .....	66
CALL OPEN .....	66
CALL CLOSE .....	67
CALL FINP .....	67
CALL FMT .....	68
CALL SYSIO .....	69
INCW / OUTCW .....	69
CALL MOVE .....	70
CALL FIC .....	70
CALL DAS .....	71
CALL CSB .....	71
CALL BIFX .....	71
CALL name .....	72
SUBROUTINE / RETURN .....	72
EXTERNAL .....	72
<b>Fehlermeldungen und Fehlercodes</b>	
Warnungen .....	73
Library-Fehler .....	73
Fehler der Analysephase .....	74
Fehler der Generierungsphase .....	75
<b>BUS</b>	
PIN-Belegung .....	76
Signal-Beschreibung .....	77
<b>CPU</b>	
Allgemeines .....	78
Blockschaltbild .....	79
Serielle Schnittstellen .....	80
<b>TASTATUR</b>	
Die Tastatur .....	81
Tastatur-Code .....	82
<b>TASTATUR INTERFACE</b>	
Technische Beschreibung .....	83
PIN-Belegung .....	84
Blockschaltbild .....	85
Funktionsbeschreibung .....	86

<b>CRT-INTERFACE</b>	Technische Beschreibung .....	Seite 87
	PIN-Belegung .....	88
	Blockschaltbild .....	89
	Funktionsbeschreibung .....	90
<b>16k RAM</b>	Technische Beschreibung .....	94
	Funktionsbeschreibung .....	94
	PIN-Belegung .....	95
	Blockschaltbild .....	96
<b>STROM - VERSORGUNG</b>	Grundmodul .....	97
	Erweiterungsmodul .....	98
<b>MINI FLOPPY IF</b>	Technische Beschreibung .....	99
	PIN-Belegung .....	100
<b>PERIPHERIE</b>	Mini-Floppy-Drive .....	101
	Disketten-Organisation .....	103
	Disketten-Handling .....	104
	Schreibkerndrucker SD4035 .....	105
	ASCII-Tabelle .....	Anhang A
	Fehlermeldungen und -Codes .....	Anhang B
	Stichwortverzeichnis .....	Anhang C





Sie haben ein flexibles, modular aufgebautes Microcomputer-System erworben, das zahlreichen Ansprüchen gerecht wird. Ob Sie das Gerät als Hobby-Computer, als Lehr- bzw. Lernsystem, als Entwicklungssystem oder für einen bestimmten Anwendungszweck benutzen wollen - alles ist möglich!

Dieses Manual soll Ihnen eine Hilfestellung bei der Inbetriebnahme sein.

Aber: Microcomputer-Technologie kann ebenso, wie die Entwicklung eigener Programme, eine packende Beschäftigung werden. In jedem Fall wird jedoch ein gewisses Engagement, sprich Lernwille, je nach Grad der Vorkenntnisse, vonnöten sein. Wir empfehlen Ihnen daher die aufmerksame Lektüre dieses Manuals.

Es würde den Rahmen dieser Broschüre sprengen, umfangreiches Microcomputer-Fachwissen vermitteln zu wollen. Für den Anwender gibt es eine ganze Palette verschiedener Literatur: Manuals, Dokumentation, bis hin zur "Teach-Easy" Lehcassette/Diskette. Eine Aufstellung dieses Angebotes finden Sie in Kapitel "Allgemeines". Zudem verweisen wir im Zuge dieses Manuals auf allgemein erhältliche Literatur, die Ihnen Ihr Fachhändler gerne beschafft.



## **EINLEITUNG** **Zu diesem Manual**

Zur Benützung dieses Manuals: Allen Ungeduldigen, die ihrem System etwas abverlangen wollen, bevor sie dieses Manual mehr oder weniger durchgearbeitet haben, empfehlen wir die Kapitel "Inbetriebnahme" und "Zum Aufwärmen".

Danach, je nach Neigung Hardware - Software, bzw. je nach Vorkenntnissen, empfehlen wir die Lektüre des Kapitels "Allgemeines". Sie kennen dann das Gerät in den Grundzügen und auch, was es kann.

Damit stecken Sie schon mitten drin ... A propos "Hardware - Software": (oder: "Hartware - Weichware"?) Ein Großteil der einschlägigen Literatur kam und kommt ebenso, wie grundsätzliche Entwicklungen, aus den USA; und damit eine Menge englischer Fachausdrücke. Wir werden diese Termini im wesentlichen auch verwenden, da Übersetzungen oftmals Begriffsverwirrung stiften und andererseits viele dieser englischen Fachausdrücke bereits die allgemeine Verständigungs-Basis bilden. Zum besseren Verständnis werden wir jedoch solche Fachausdrücke in notwendigem Maß erklären.

**Hardware**, der materielle oder „harte“ Teil eines Computersystems. In der Hauptsache aktive oder passive Schaltelemente, Peripherie etc.

**Software**, der immaterielle Programmanteil eines Computersystems. Erst Software (Programme) läßt die Hardware programmierte Abläufe abwickeln.



## ALLGEMEINES Systembeschreibung

Das System ist modular aufgebaut. Das heißt, der mechanische und elektrische Aufbau ist so konzipiert, daß verschiedene Konfigurationen je nach Anwendungsfall zusammengestellt werden können. Das Herz stellt ein 19-Zoll Einschubrahmen dar, der die verschiedenen zentralen Hardware-Komponenten aufnimmt. Verschiedene Leiterplatten (Print-Platinen) können in Abhängigkeit des Anwendungszweckes eingeschoben werden. Alle Leiterplatten sind 100 x 160 mm groß. Dieses sog. Europakarten-Format ist ebenso wie die 19-Zoll-Einschubtechnik, Industriestandard.

Einige dieser Leiterplatten sind für die Funktion unbedingt notwendig (Minimalkonfiguration), andere dienen zur Erweiterung für bestimmte Anwendungsfälle, bzw. zur Kommunikation mit der Außenwelt. Die elektrische Verbindung der einzelnen Leiterplatten geschieht über die "Bus-Platine". Über 96-polige Steckverbindungen werden die einzelnen Europa-Platinen auf die Bus-Platine aufgesteckt und so die elektrische Verbindung untereinander hergestellt. Das Bus-System selbst benötigt nur 64 Leitungen. Die restlichen 32 Kontakte der 96-poligen Steckverbindung dienen speziellen Zwecken (im Normalfall Verbindungsleitungen zu verschiedenen Bediengeräten).

Auf der Rückseite der Bus-Platine sind außerdem die auch von außen erkennbaren Steckverbindungen für den Anschluß externer Geräte eingelötet. Solche Geräte können z. B. Drucker, Cassetten-Recorder, Fernseher oder Monitor sein.

Im 19-Zoll Einschubrahmen befinden sich auch die notwendigen Netzteile, die Ihr Gerät und eventuelle Erweiterungen mit den notwendigen Spannungen versorgen.

Insgesamt bietet die Bus-Platine Platz zum Einschub für 9 Europa-Platinen. Ventilatoren laufen, sobald Sie Ihren Computer einschalten.

Damit Sie Ihrem System auch irgendwie mitteilen können, was Sie von ihm wollen, besitzt es eine Tastatur in professioneller Ausführung mit eigenem "Zehner-Block". Und weil Sie sehen wollen, ob das System das, was Sie von ihm wollten, dann auch wirklich getan hat, schließen Sie einfach Ihren Fernseher an. Für höhere Ansprüche können Sie auch einen Industrie-Monitor anschließen. Ihr Händler berät Sie gerne.

Bus, ein Informationweg, über den Daten verschiedener Quellen zu verschiedenen Zielorten übertragen werden. In der Computertechnik meist Leiterbahnen aus Kupfer, die auf eine Kunststoff-Platte kaschiert sind (Bus-Platine im engl. oft: „Mother-Board“).



alphaTronic

## **ALLGEMEINES Expansions-Möglichkeiten**

### **Expansion**

Irgendeinmal liegt die Zeit, in der Sie Ihr erstes eigenes Programm zum Laufen brachten, weit hinter Ihnen. Sie kennen dann Ihr System und dürfen mit Fug und Recht behaupten, Programmierer zu sein.

Der Sinn steht Ihnen nach Höherem. Vielleicht wollen Sie Ihre Modelleisenbahn steuern, die eigene Buchhaltung erledigen (und womöglich noch für ein paar Freunde) oder ...

Dann ist der Zeitpunkt gekommen, das System genau Ihren Anforderungen anzupassen, ohne daß eine bereits vorhandene Komponente dadurch wertlos wird. Eine Liste der Expansionsmöglichkeiten, die zudem laufend erweitert wird, steht zur Verfügung.

---

### **Expansions-Möglichkeiten**

Fernsehgerät (16 x 32 Zeichen)  
Fernsehgerät (16 x 64 Zeichen)  
12 Zoll Monitor (hochauflösend mit 16 x 16 Matrix,  
24 x 80 Zeichen)  
16 KB RAM Speichererweiterung  
Echtzeituhrplatine  
ECMA-34, Cassettenlaufwerk V 24-Version, max.  
150 kB  
Video Controller mit Farbzusatz-Platine  
Hobby Parallel-Interface  
Cassetten-Recorder  
Metallpapier-Drucker  
Schreibkerndrucker SD4035  
2 Mini Floppy Laufwerke

Jeder Computer braucht, wenn er etwas tun soll, Software!

Das wissen wir bereits, und wir wissen schon etwas über die Hardware-Möglichkeiten. Sicher haben Sie Ihr Gerät schon eingeschaltet. (Siehe "Inbetriebnahme"). Wenn alles richtig angeschlossen ist, erscheinen am Bildschirm bereits einige Textzeilen und in der linken unteren Ecke ein weißes Viereck, der Cursor. Ohne im Augenblick näher darauf einzugehen, was dieser Text bedeutet, interessiert uns im Moment nur folgende Feststellung: Offensichtlich passiert in Ihrem Computer bereits etwas. Da das aber nur mit Hilfe von Software, also eines Programmes möglich sein kann, muß in Ihrem Computer bereits Software vorhanden sein.

Das heißt: Obwohl Sie als Amateur noch keinerlei Software in den Computer geladen haben, laufen in Ihrem System bereits Programme ab. Das ist in der Tat so. Um welche Programme, welche System-Software es sich dabei handelt, ist abhängig von der Version, die Sie besitzen. Nachstehend finden Sie eine Übersicht, welche System-Software in welchen Versionen als Standard vorhanden ist, bzw. mitgeliefert wird.

Version	System Software
1.2	MOS-Monitor BASIC-Interpreter
1.3	MOS-Monitor BASIC-Compiler
2	MOS-Monitor BASIC-Compiler (auf Diskette)
3	MOS-Monitor BASIC-Compiler (auf Diskette)

**System-Software.** Hierunter versteht man Programme, die zum Betrieb des Computers erforderlich sind und in der Regel vom Hersteller zur Verfügung gestellt werden, z. B.

- Betriebssysteme,
- Übersetzungsprogramme,
- Dienstprogramme

**Basic-Interpreter.** Dies ist ein Programm, das eine Anweisung (Basic-Befehl) interpretiert, in die Maschinensprache übersetzt, ausführt, den nächsten Befehl interpretiert usw.

**Basic-Compiler,** ein Programm zum Übersetzen der Basic-Befehle in die Maschinensprache einer bestimmten Datenverarbeitungsanlage.

**MOS-Monitor.** Als Microcomputer-Operating-System wird das Betriebssystem des KISS-Computers bezeichnet. Wie jedes Betriebssystem besteht auch das MOS aus einer Reihe von Programmen zur Abwicklung und Verwaltung von Anwenderprogrammen, Ein-/Ausgabeverwaltung, Behandlung der Programmunterbrechungen, Verwalten des Speichers, usw.

Außer dieser im Lieferumfang enthaltenen System-Software ist, je nach Ausbaustufe und Bedarf, weitere System-Software erhältlich:

- 8080 Assembler
- Editor
- Disassembler
- Mikbug-Testmonitor
- Promcode
- Formatier-, Kopier- und Vergleichsprogramm



Soweit eine Kurzdarstellung der Systemsoftware-Möglichkeiten. Eine Einführung in die Standard-System-Software finden Sie in diesem Manual in den entsprechenden Kapiteln. Darüber hinaus gibt es für jede Programmiersprache noch ein ausführliches Manual. Bitte beachten Sie die entsprechenden Hinweise.

System-Software dient dem Benutzer eines Computersystems als Verständigungsbasis, um dem Computer verschiedene Anweisungen (Befehle) zu geben, also genau das auszuführen, was der Anwender programmiert hat. Das kann mehr oder weniger aufwendig erfolgen, auf verschiedenen Ebenen unterschiedlicher Programmiersprachen. Eine Darstellung dieser Zusammenhänge finden Sie in den Kapiteln "MOS-Monitor" und "Basic-Compiler".

Wenn Sie sich also dieser System-Software bedienen, einer Programmiersprache im weitesten Sinn, wird Ihr System die Befehle ausführen, die Sie ihm eingeben. Eine mehr oder weniger lange Folge solcher Befehle ist dann ein Programm. Ein Programm also, dessen Zweck und Ablauf vom Anwender bestimmt wird. Solche Anwender-Programme kann man für die verschiedensten Zwecke auch kaufen.

Fragen Sie Ihren Fachhändler!

Für das System steht zur Softwareerstellung ein BASIC-Compiler zur Verfügung. Die Details hierzu finden Sie in dem entsprechenden Kapitel.

Für das Selbststudium der Programmiersprache BASIC (Beginners of all Symbolic Instruction) empfehlen wir Ihnen folgende Bücher:

**BASIC für Anfänger**

Dieses Buch ist für Anfänger auf dem Gebiet der Datenverarbeitung geschrieben. Besondere mathematische Kenntnisse werden nicht vorausgesetzt. An Hand von vollständig durchgeführten Musterbeispielen dringt der Leser schrittweise in das Wissensgebiet ein. Viele Übungen mit angegebenen Lösungen geben ihm die Möglichkeit, die erworbenen Kenntnisse zu überprüfen.

Oldenbourg Verlag, 155 Seiten

**Anleitung zum praktischen Gebrauch von BASIC**

Es ist die Absicht dieses Buches, den Leser zum praktischen Gebrauch der Programmiersprache BASIC zu führen. Der Text berücksichtigt sowohl kommerzielle wie technisch-wissenschaftliche Probleme. In einem Anhang wird ein Vergleich der unterschiedlichsten Versionen von BASIC gegeben.

Oldenbourg Verlag, 240 Seiten

**Mikroprozessor Software Handbuch MSH**

Grundlagen und Einführung in die Programmierung von Mikrocomputern in den wichtigsten Programmiersprachen (Basic, Fortran, Assembler). Zusammenstellung der wichtigsten Befehlslisten: 8080, Z80, M6800, etc. Ein Software-Handbuch für jeden, der mit Mikrocomputern zu tun hat.

Ing. W. Hofacker Verlag, ca. 200 Seiten

**57 Programme in BASIC**

Ein Buch mit technisch-wissenschaftlichen Programmen und einer großen Anzahl von Spielprogrammen in Basic. Ein Buch für jeden, der sich mit der Mikrocomputertechnik befassen will. Alle Listings sind in Basic und können auf den meisten Mikrocomputer-System gefahren werden.

Ing. W. Hofacker Verlag



alphaTronic

## VARIANTEN Konfigurations-Möglichkeiten

Auf Grund des modularen Aufbaus ist es dem Anwender möglich, eine Systemkonfiguration seines Systems ganz nach Wunsch vorzunehmen. Es ist durchaus möglich, zunächst mit einer relativ bescheidenen Konfiguration zu beginnen und das System im Laufe der Zeit je nach Möglichkeit entsprechend auszubauen. In der Folge ein möglicher Ausbauvorschlag in Stufen, der, den persönlichen Bedürfnissen entsprechend, modifiziert werden kann.

Da das System auf dem MC 80 Bus-System aufgebaut ist, können über eine entsprechende Bus-Erweiterung sämtliche MC-80 Bus-Platinen des Herstellers verwendet werden.

### 1. Stufe:

Mit Anschlußbuchsen für handelsüblichen Cassettenrecorder und Fernsehgerät komplett mit Tastatur.

### 2. Stufe:

Mini-Floppy-Interface und 1-2 Mini-Floppy-Laufwerke.

### 3. Stufe:

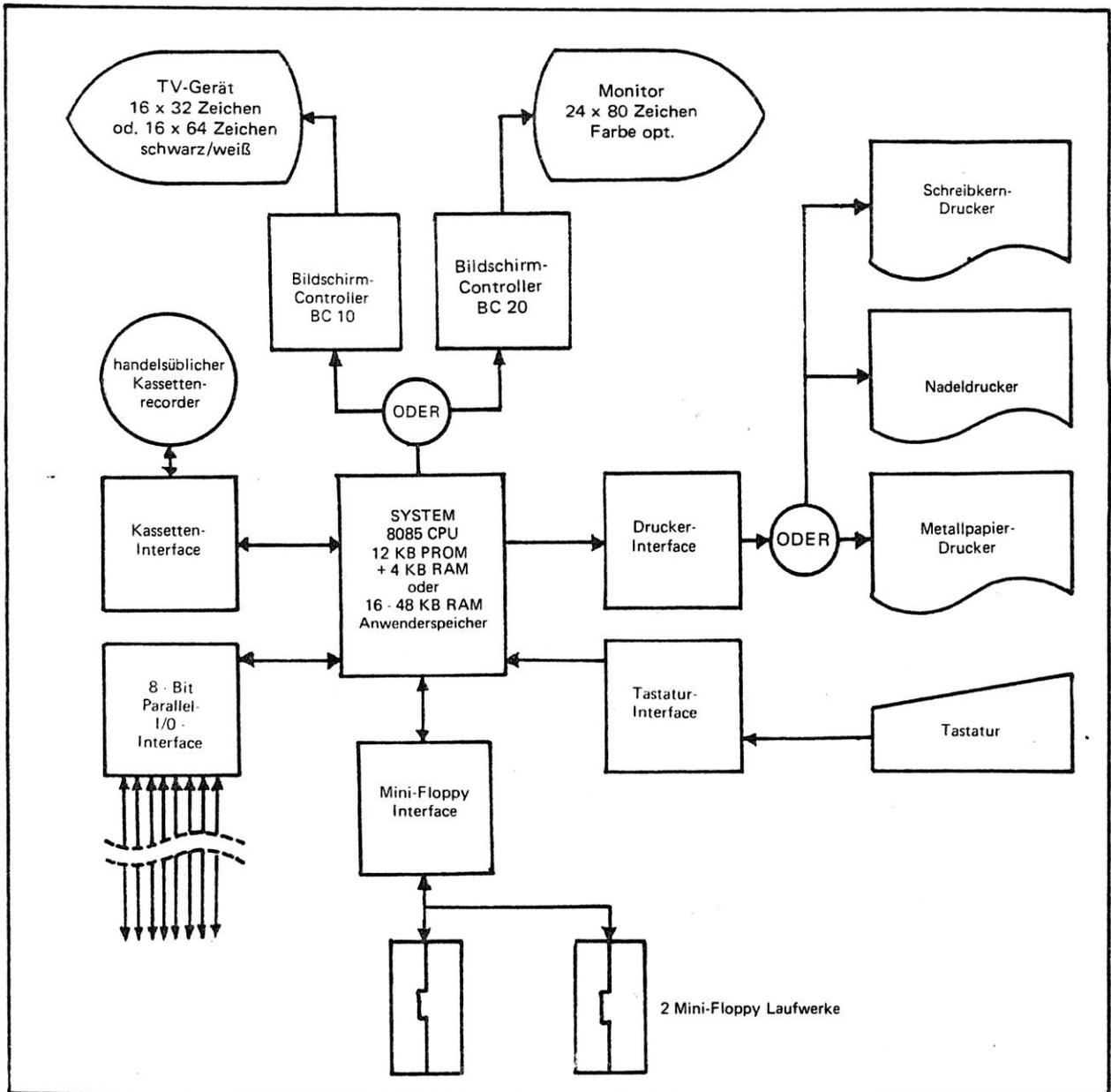
Professioneller 12 Zoll Video-Monitor mit 24 x 80 Zeichen.

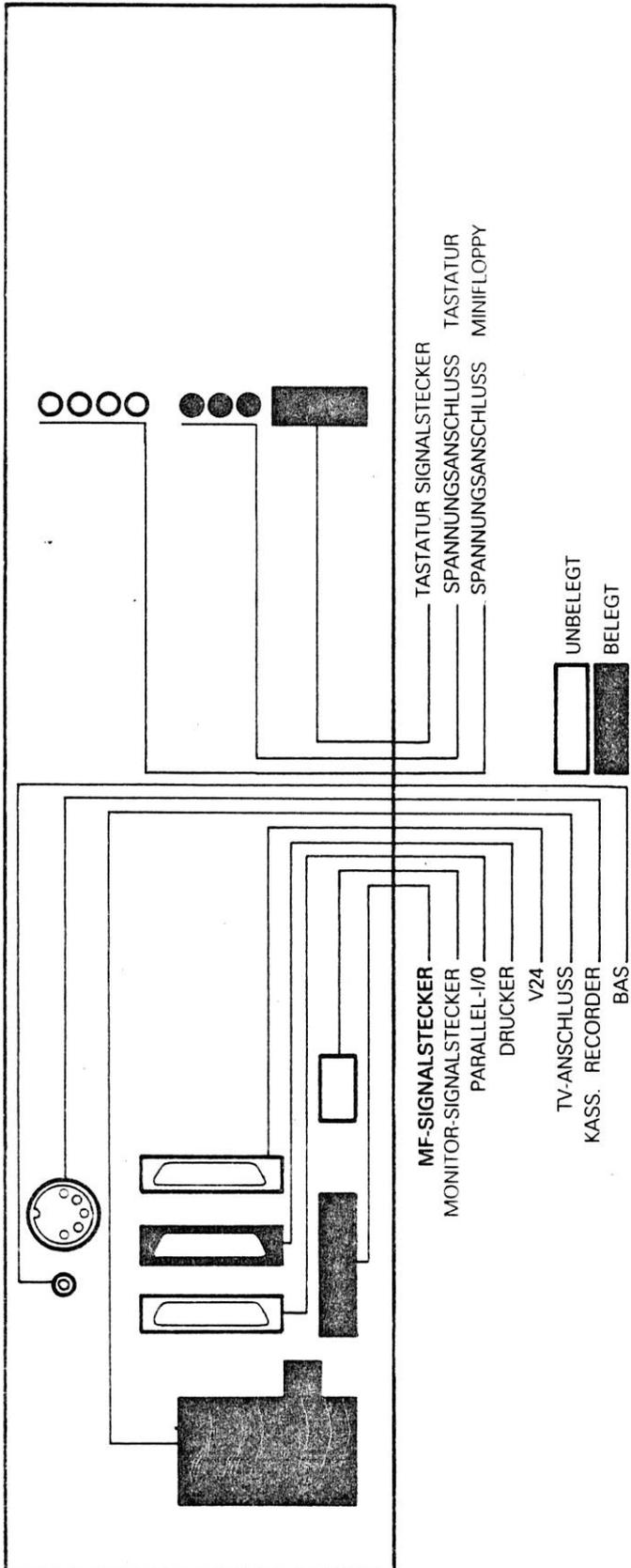
### 4. Stufe:

Speichererweiterungen nach Wunsch.

Ein so ausgerüstetes System läßt, als Beispiel gesehen, kaum mehr Wünsche offen. Natürlich sind auch grundsätzlich andere Zusammenstellungen möglich: Ein auf Cassettenrecordern, eventuell in professioneller Ausführung, oder für Farbe über Ihren Farbfernseher ausgerüstetes System ist ebenso denkbar, wie die Applikation von 8 Zoll Disketten. Außerdem stehen verschiedene Druckerversionen zur Verfügung.

**Interface.** Als Interface, Schnittstelle oder Nahtstelle wird die Verbindung zwischen den einzelnen Elementen eines Computers bezeichnet. Z. B. zwischen der CPU und dem Floppy-Laufwerk oder dem Drucker ist jeweils ein Interface erforderlich.

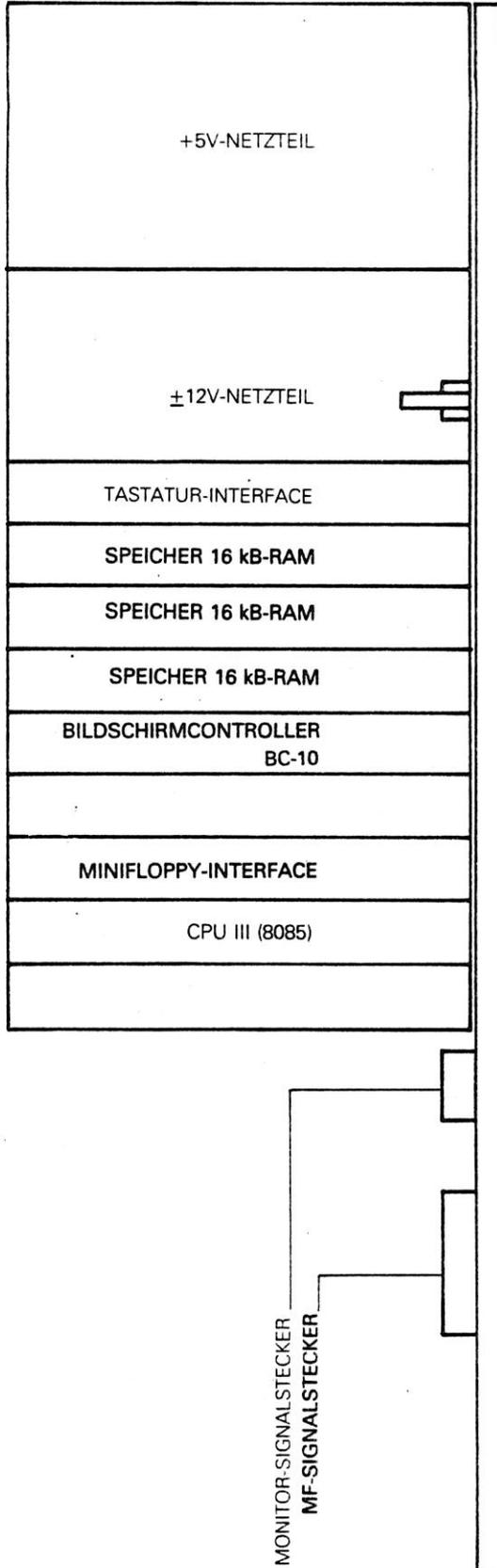






# VARIANTEN Baugruppenordnung

**System-SW**

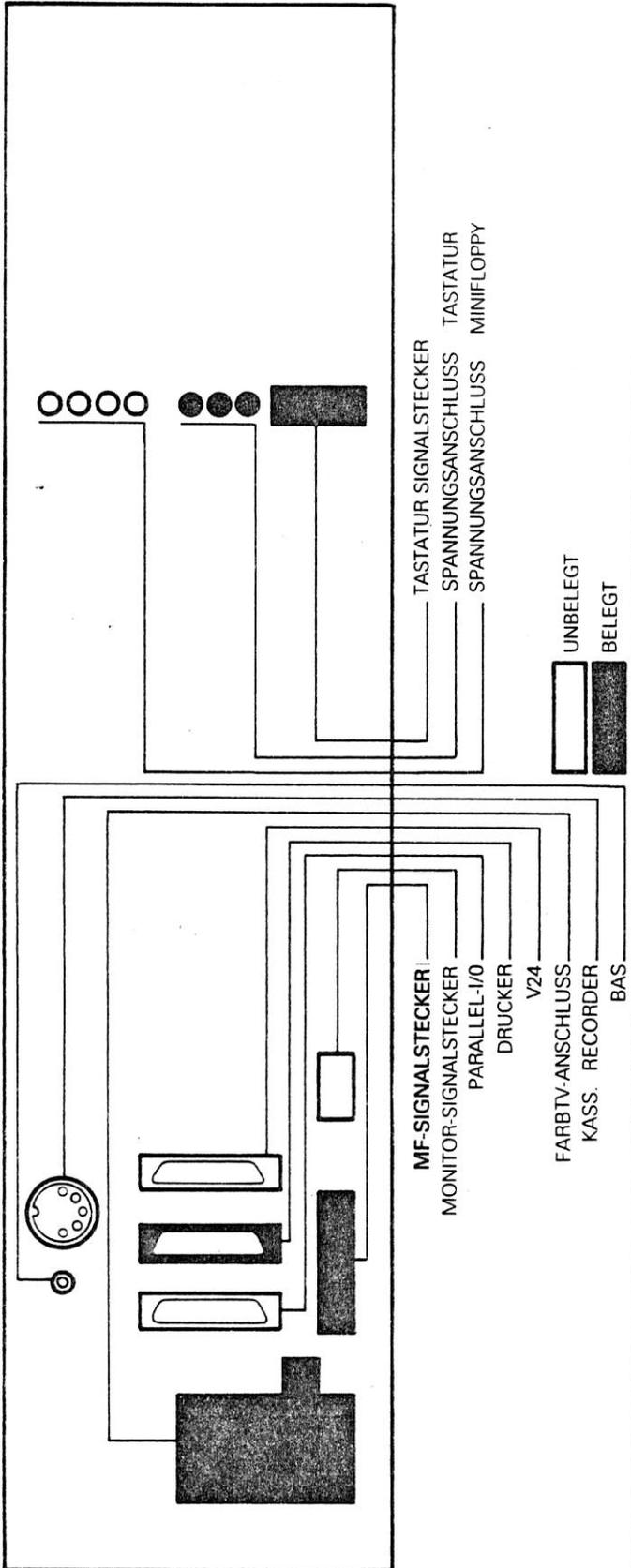




# VARIANTEN

## Baugruppenordnung

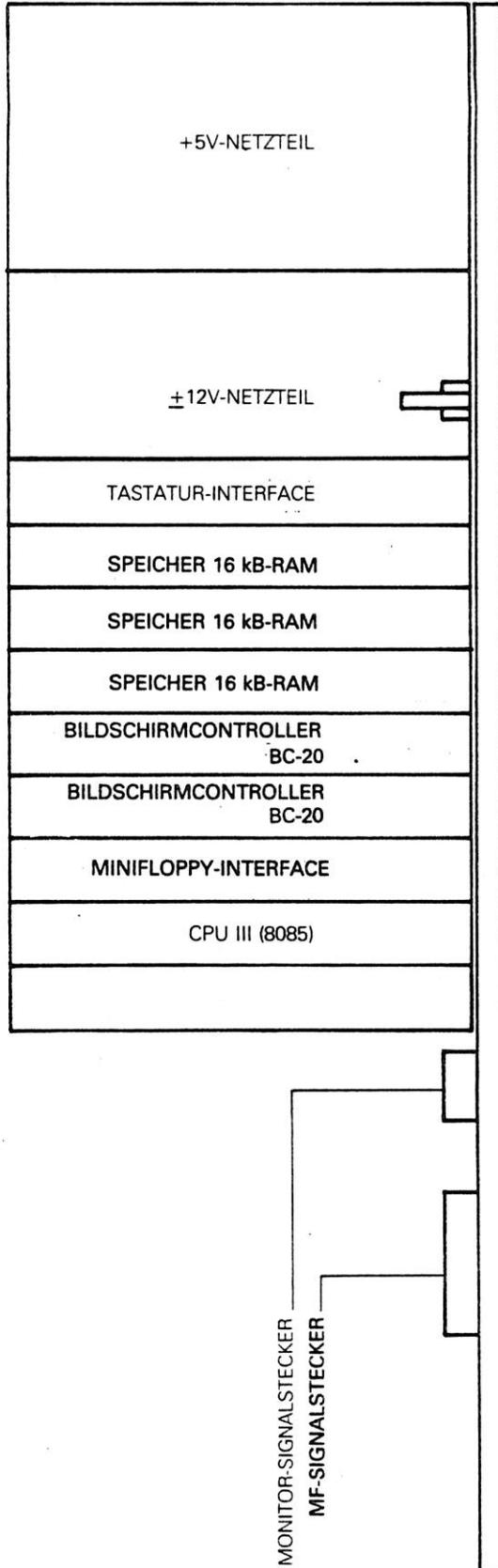
**System-Farbe**





# VARIANTEN Baugruppenordnung

System-Farbe



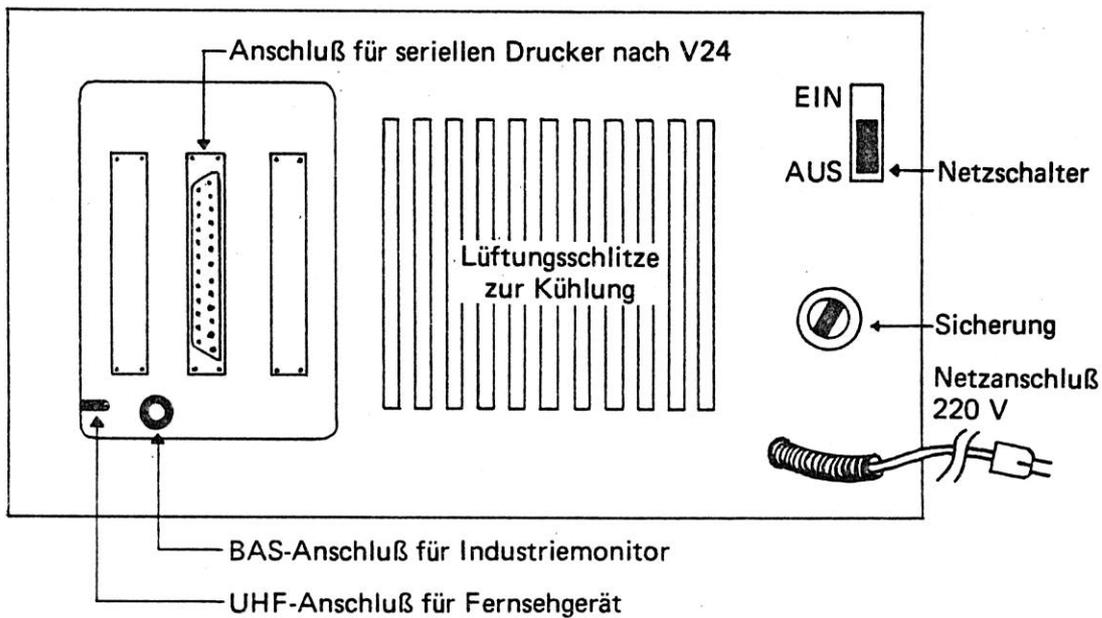
Jetzt ist es so weit!

Nachdem Sie Ihren Computer ausgepackt haben, fiebern Sie sicher schon darauf auszuprobieren, was er kann. Trotzdem empfehlen wir Ihnen, sich ausführlich mit diesem und weiteren Manuals, bzw. Fachliteratur zu beschäftigen.

Für alle, denen das zu lange dauert, haben wir im folgenden Kapitel eine Kurzfassung für die erste Inbetriebnahme zusammengestellt. Zunächst jedoch versorgen Sie Ihr Gerät mit den entsprechenden Anschlüssen.

Je nach Ausbaustufe, für die Sie sich entschieden haben, sind verschiedene Peripheriegeräte anzuschließen. Auf der Rückseite des Gerätes befinden sich verschiedene Anschlußbuchsen. Diese sind in Abhängigkeit Ihrer Version entweder beschaltet oder für zukünftige Erweiterungen vorbereitet.

**Achtung:** Bevor Sie periphere Geräte an Ihr System anschließen, schalten Sie es ab und ziehen Sie den Netzstecker.



Sie können an Ihr System, je nach Ausstattung, mehrere Bildschirme anschließen.

### 1. Anschluß eines handelsüblichen schwarz/weiß-Fernsehgerätes

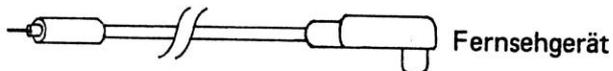
Voraussetzung:

- Bildschirm-Controller BC 10 (Standardausstattung)
- Antennenanschlußkabel (wird mitgeliefert)

Vorgang:

Stecken Sie das Antennenkabel mit dem kleineren Stecker in die Buchse "UHG-Ausgang" an Ihr Gerät und den größeren Stecker des Kabels in den Antennen-eingang Ihres Fernsehgerätes.

Stellen Sie das Fernsehgerät auf UHF, Kanal 32, ein.



### 2. Anschluß eines handelsüblichen Farb-Fernsehgerätes

Voraussetzung:

- Bildschirm-Controller BC 20 (Optionelle Ausstattung)
- Antennenanschlußkabel (wird mitgeliefert)

Vorgang:

Siehe 1.

### 3. Anschluß eines Industriemonitors

Voraussetzung:

- Bildschirm-Controller BC 10 oder BC 20
- Anschlußkabel (wird mitgeliefert)

Vorgang:

Stecken Sie das Anschlußkabel mit dem runden Stecker in den sog. "BAS-Ausgang" an Ihrem Gerät und den flachen Stecker des Kabels in den Industriemonitor.



An das System lassen sich mehrere Drucker anschließen. Sie können entsprechend Ihren Bedürfnissen wählen, zwischen

- Metallpapierdrucker
- Matrixdrucker
- Schreibkerndrucker

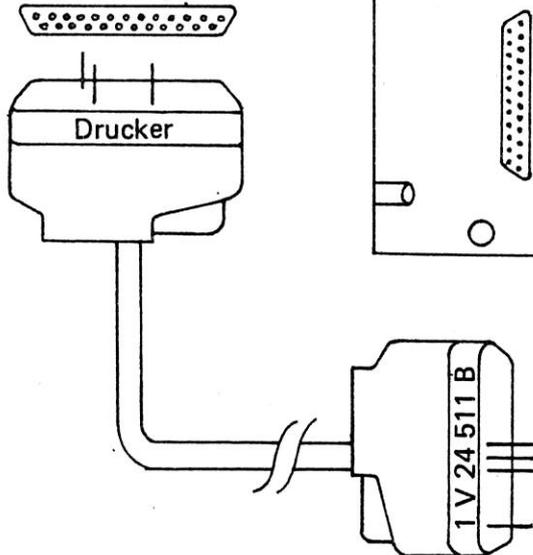
Alle Drucker werden bereits mit der entsprechenden Anpassung (serielle Schnittstelle nach RS 232) und einem passenden Anschlußkabel geliefert.

Vorgang:

- Schließen Sie den Stecker mit der Aufschrift "Drucker" beim jeweiligen Drucker an.
- Stecken Sie den Stecker mit der Aufschrift „V 24 511 B“ in das Gerät.

### Drucker -Anschluß

Steckdose am Drucker  
25 Anschlußpunkte



Anschluß-  
steckdose  
für serielle  
Drucker n. V 24



alphatronic

# INBETRIEBNAHME

## Reset

Unter Reset (Zurücksetzen) versteht man einen Vorgang, bei dem – ausgelöst durch einen Reset-Impuls – ein bestimmtes Initialisierungsprogramm abläuft. Dieses Initialisierungsprogramm ist im Monitor enthalten. Ausgelöst werden kann dieser Reset-Impuls auf zwei Arten:

Einmal durch den sogenannten Power-on Reset, nach Einschalten der Netzspannung, und zum zweiten durch Drücken der Reset-Taste (rechte obere Taste des Tastenfeldes mit der Beschriftung "Ein").

Grundsätzlich erfolgt nach Auslösung des Reset-Impulses bei dem verwendeten Mikroprozessor 8085 ein Programmsprung zur Adresse 0000. An dieser Adresse in Ihrem Speicher beginnt der Ablauf des MOS-Programmes (Mikrocomputer-Operating-System Monitor). Das ist ein in Ihrem System fest eingespeichertes Programm. Beginnend an Speicheradresse 0000, werden in der Folge nach dem Reset bestimmte Programmsequenzen durchlaufen, die eine Initialisierung bewirken.

Dabei handelt es sich beim Power-on Reset oder beim normalen Reset (Reset-Taste) um einen sog. Kaltstart. Im Prinzip werden dabei verschiedene Register gesetzt und Variablen eingestellt, die für die weitere Benutzung Ihres Computers notwendig sind. Über den MOS-Monitor gibt es ein entsprechendes Manual, in dem die genauen Vorgänge beschrieben sind.

Der MOS-Monitor meldet sich auf Ihrem Bildschirm mit vier Zeilen, zum Beispiel:

```
RESET
MOS 4/F V.3.1
$ 1976 4010 FFEF;
```



Cursor

Unter dem MOS-Monitor sind alle Befehle, beginnend mit „.“ einzugeben.

**Initialisierung.** Etwas Vorbereitetes tatsächlich in Gang bringen oder veranlassen.

*Einschalt*

**Register** ist ein funktionell oder konstruktiv ausgezeichnetes Speicherelement für ein Datenwort oder eine Speicheradresse. Physikalisch gesehen, kann es im Prozessor oder Speicher realisiert sein. Wichtige Register stellen z. B. das Indexregister, der Akkumulator, das Unterbrechungsregister, das Befehlsregister und das Adreßregister dar.

Variantenbezeichnung

**Cursor.** Positionsanzeiger, um bei Bildschirmterminals die augenblickliche Position anzuzeigen. Ein eingegebenes Zeichen erscheint dann an dieser Position und der Cursor wandert um 1 Stelle weiter.

Ihr System ist jetzt bereit zum Empfang von MOS-Befehlen.



## INBETRIEBNAHME Initialisieren

Die Versionen "2 . ." aufwärts sind standardmäßig mit einem Mini-Floppy-Laufwerk ausgerüstet. Der wesentliche Unterschied zu den Versionen "1 . ." besteht darin, daß in diesen, mit Mini-Floppy-Laufwerken ausgerüsteten Systemen in den Festspeichern nur das MOS-Monitor-Betriebssystem fest eingespeichert ist. Zahlreiche andere Programmiersprachen und Anwenderprogramme können von der Mini-Floppy in den Computer eingelesen werden. Serienmäßig mitgeliefert werden zwei Disketten: Die 1. Diskette enthält den BASIC-Compiler, die 2. Diskette enthält das Formatier-, Kopier- und Vergleichsprogramm.

Darüber hinaus sind zur Zeit weitere Disketten lieferbar:

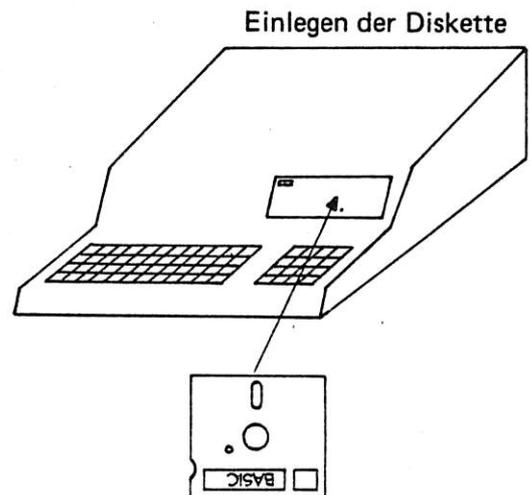
8080 Assembler  
8080 Dis-Assembler  
Mikbug  
Editor  
und Promcode.

Des Weiteren steht noch eine Anzahl von Anwenderprogrammen auf Diskette zur Verfügung. Dieses Sortiment wird laufend erweitert. Fragen Sie Ihren Fachhändler.

Nachdem Sie Ihr Gerät mit Mini-Floppy eingeschaltet haben, sollte eine Ihrer ersten Arbeiten unbedingt darin bestehen, die mitgelieferten Systemdisketten (BASIC-Compiler und Formatierungsprogramm) auf weitere Disketten zu duplizieren.

Disketten sind in dieser Preisklasse das modernste und wohl am meisten verbreitete Massenspeichermedium. Wir empfehlen Ihnen, die Disketten schonungsvoll zu behandeln: Stecken Sie Ihre Disketten nach Gebrauch immer in die mitgelieferten Schutzhüllen, knicken Sie sie nicht, und schützen Sie sie vor übermäßiger Wärme und mechanischer Beanspruchung. Wenn Sie Ihre Disketten beschriften, tun Sie dies niemals mit Kugelschreiber oder Bleistift, sondern verwenden Sie dazu weiche Filzstifte, um ein Durchprägen durch die Schutzhülle auf die Diskette zu verhindern.

Trotzdem sollten Sie von allen Disketten, die für Sie wichtige Daten enthalten, unbedingt mindestens ein Duplikat besitzen. Das gilt natürlich auch für die mitgelieferten System-Disketten. Wie das geschieht, entnehmen Sie bitte den folgenden Seiten.





## **INBETRIEBNAHME** **Formatieren Diskette**

Fabrikneue Disketten müssen vor der ersten Inbetriebnahme formatiert werden. Unter Formatieren versteht man das Einteilen der Diskette in eine bestimmte Anzahl von Spuren und Sektoren (ähnlich Tortenecken), die es ermöglichen, Information auf die Diskette abzuspeichern, und vor allem, sie zu einem späteren Zeitpunkt wieder zu finden.

Achtung: Formatieren kann man auch bereits beschriebene Disketten!

Beachten Sie dabei aber, daß die darauf enthaltene Information für Sie verloren ist!

Das Formatieren geschieht mit dem Formatier- und Kopierprogramm. Dieses Programm befindet sich auf der mitgelieferten System-Diskette 2. Der Formatierungsvorgang einer oder mehrerer Disketten läuft wie folgt ab:

1. Nachdem Sie Ihr Gerät eingeschaltet haben, meldet sich der MOS-Monitor, siehe S. 17
2. Legen Sie die System-Diskette mit dem Formatier- und Kopierprogramm ein.
3. Geben Sie . B ein und drücken Sie die innere SP-Taste.
4. Das Programm meldet sich wie folgt:  
\*FLOPPY FORMATIEREN UND KOPIEREN\*  
FORMATIEREN → F  
KOPIEREN → K  
> ■
5. Geben Sie F ein und drücken Sie die innere SP-Taste. Am Bildschirm erscheint:  
FORMATIEREN:  
FLOPPY EINLEGEN: START → S  
> ■
6. Legen Sie die zu formatierende Diskette (Schreibschutz beachten, siehe Anmerkung) ein, geben Sie über die Tastatur S ein und drücken Sie die innere SP-Taste.
7. Das Programm formatiert die gesamte Diskette und meldet sich:  
OPERATIONSENDE: NEUSTART → N
8. Möchten Sie noch weitere Disketten formatieren, so geben Sie bitte N ein. Weiter bei 4. Andernfalls drücken Sie die RESET-Taste und der MOS-Monitor meldet sich wieder.

Anmerkung:

Mini-Disketten besitzen eine seitliche Ausstanzung, eine Kerbe, die einen Schreibschutz bedeutet. Ist diese Kerbe offen, so kann die entsprechende Diskette nicht be- bzw. überschrieben werden.

Soll eine Diskette beschrieben (auch formatiert) werden, so ist die Schreibschutzkerbe durch einen Aufkleber abzudecken.



alphaTronic

## INBETRIEBNAHME

Duplizieren Systemdiskette

Das Duplizieren der Systemdisketten sollte eine der ersten Arbeiten sein, die Sie mit Ihrem Computer durchführen, um sich im Falle der Zerstörung der Systemdiskette nicht der Gefahr auszusetzen, keinen Ersatz dafür zu haben.

Grundsätzlich: Disketten (übrigens auch Cassetten) werden immer dupliziert, indem man den Inhalt des Originals von der Diskette in den Arbeitsspeicher lädt und von dort wieder auf die Duplikat-Diskette schreibt.

Die Diskettenorganisation ist sequentiell File-orientiert. Als File bezeichnet man einen in sich abgeschlossenen Datenbestand. Das können Daten verschiedenster Art sein: Programm, Namenslisten, Telefonnummern, Datenbestände allgemeiner Art etc.

Für Ihren Computer heißt das: Auf Disketten können verschiedene Datenbestände (also auch Programme) in sequentieller Reihenfolge abgespeichert werden.

Im einzelnen läuft das Duplizieren von Disketten wie folgt ab:

1. Laden des Formatier- und Kopierprogrammes wie auf Seite 19, Punkt 1 - 4 beschrieben.
2. > K eingeben  
KOPIEREN:  
MIT 1 ODER 2 LAUFWERKEN? → 1/2  
> ■
3. 1 eingeben  
QUELL-FLOPPY EINLEGEN: START → S  
> ■
4. S eingeben  
Bis Spur 20 wird die Diskette eingelesen  
ZIEL-FLOPPY EINLEGEN: START → S  
> ■
5. Diskette wechseln, C-Taste drücken  
S eingeben  
Bis Spur 20 wird die neue Diskette beschrieben  
QUELL-FLOPPY EINLEGEN: START → S  
> ■
6. Diskette wechseln, C-Taste drücken  
S eingeben  
Von Spur 21 - 40 wird die Diskette gelesen  
ZIEL-FLOPPY EINLEGEN: START → S  
> ■
7. Diskette wechseln, C-Taste drücken  
S eingeben  
Von Spur 21 - 40 wird die Diskette beschrieben
8. OPERATIONS-Ende: NEUSTART → N  
Siehe Seite 19

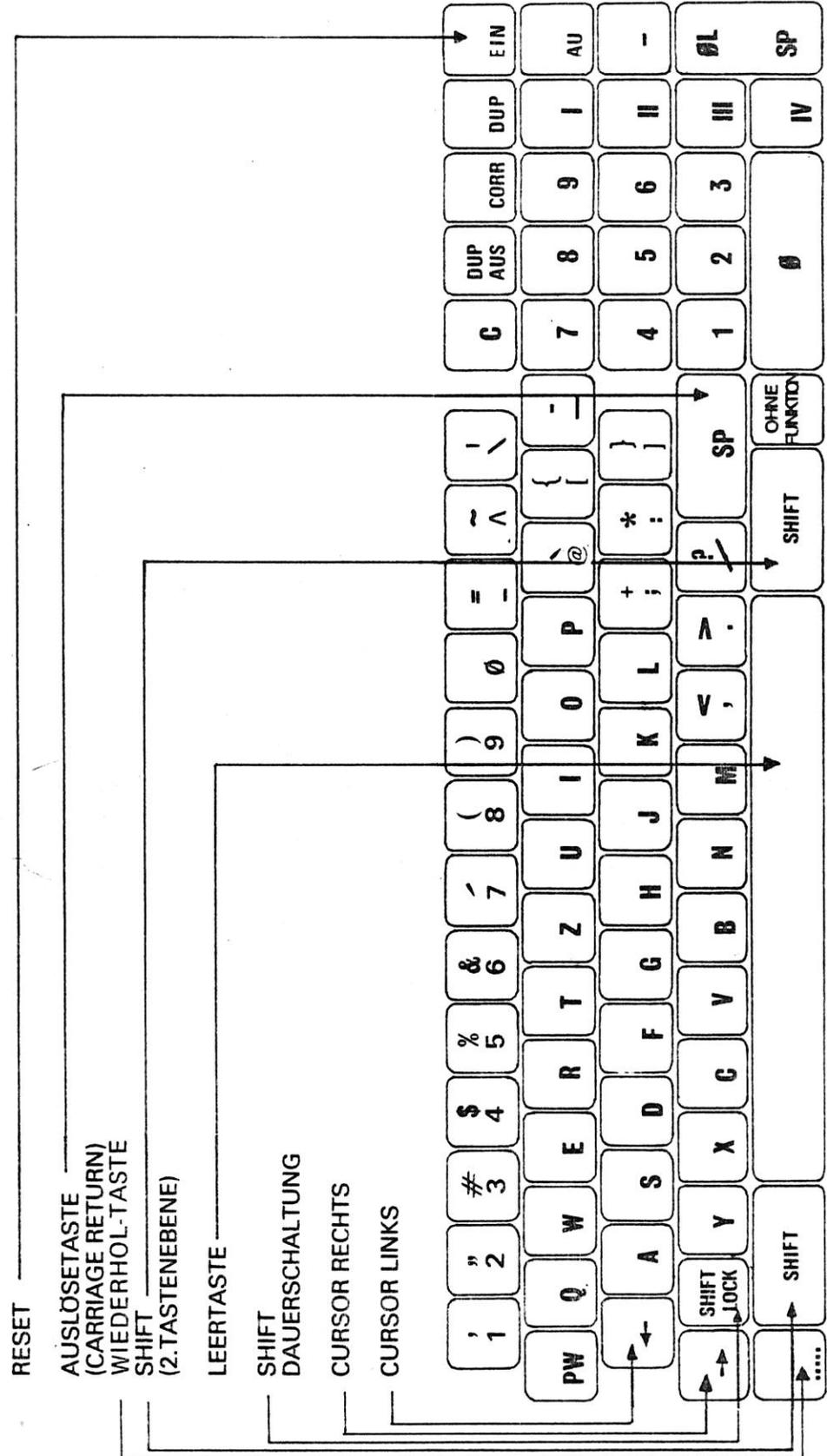


alphatronic

# ZUM AUFWÄRMEN

## Tasten-Funktionen

DUP  
Die Tasten C, AUS, CORR, DUP, AU, -, I - IV und PW stellen sog. frei-programmierbare Funktionstasten dar, d. h. diese Tasten können vom jeweiligen Anwenderprogramm abgefragt werden. Siehe Tastatur-Code.





alphaTronic

# ZUM AUFWÄRMEN

## Laden eines Programmes

Wir wissen mittlerweile, daß sich ein Programm, nämlich das MOS-Monitor-Programm, bereits im System befindet. Dieses Programm ist es, das es Ihnen sozusagen überhaupt erst ermöglicht, mit Ihrem Gerät Kontakt aufzunehmen. Außerdem versorgt es bestimmte Basisbedürfnisse wie z. B.: auf gedrückte Tasten reagieren und richtig verarbeiten oder Zeichen auf den Bildschirm schreiben, oder Daten von der Diskette in den Computer lesen und umgekehrt, usw. Damit der Computer irgendwelche Handlungen vollführen kann, benötigt er Information. Diese Information muß ihm in elektrisch verwertbarer Form zur Verfügung stehen. Das heißt: jedes Zeichen, das Sie Ihrem Computer eingeben, muß er sich merken können. Dazu dienen die sogenannten "Speicher".

Für jedes eingegebene Zeichen benötigt Ihr Computer einen Speicherplatz. Insgesamt kann er über 64.000 Speicherplätze verfügen. Ein Speicherplatz kann ein Zeichen aufnehmen, das entspricht einem Byte.

Wenn Ihr Computer also Information an Speicherplätze ablegt, muß er wissen, wohin. Zu diesem Zweck sind die Speicherplätze gewissermaßen numeriert. Nicht um die Sache zu verkomplizieren, sondern aus technischen Gründen, werden diese Speicherplätze nicht im Dezimal-System, sondern im Hexadezimal-System numeriert. Zur Lokalisierung eines bestimmten Speicherplatzes dient also eine hexadezimale Zahl – die sogenannte Speicheradresse. Der eingebaute Mikroprozessor 8085 ist in der Lage, 64 KILO BYTE (KB) Speicherbereich zu adressieren.

Da diese Speicherplätze in der Lage sein müssen, einmal dieses Programm und ein anderes mal jenes Programm abspeichern zu können, müssen sie veränderbar sein.

Diese Veränderung geschieht durch elektrische Vorgänge. Das heißt aber auch – wenn Sie Ihr Gerät vom Netz abschalten – verändert sich auch der Speicherinhalt. Da die in solchen Speichern abgelegte Information in diesem Fall verloren ist, spricht man von sogenannten "flüchtigen Speichern" oder kurz RAM (Randomaccess-Memory).

Es ist nicht immer sinnvoll, sämtliche Informationen, die im Speicher eines Computers abgelegt sind, zu verlieren, wenn der Strom abgeschaltet wird. Tatsächlich haben wir ja bereits festgestellt, daß nach dem Einschalten der Spannungsversorgung ein bestimmtes Programm abläuft. Auch dieses Programm, das MOS-Monitor-Programm, ist in Speichern abgelegt. Im Grunde handelt es sich dabei um ganz normale Speicherplätze, wie vorher beschrieben, jedoch mit dem Unterschied, daß ein einmal hineingeschriebener Inhalt für immer bestehen bleibt. Solche Speicher werden als "nicht flüchtige Speicher" oder ROM (read only Memory) bezeichnet.

**Speicher.** Alle Medien, die Informationen über einen beliebig langen Zeitraum hinweg festhalten und bei Bedarf wieder abgeben können, werden als Speicher bezeichnet. In der Datenverarbeitung werden Speicher grob in zwei Gruppen eingeteilt:

- a) Speicher, die eine Information nur einmal aufnehmen können, z. B. Lochkarten, Lochstreifen u.ä.
- b) Speicher, die immer wieder neue Informationen aufnehmen. Dies sind z. B. alle Magnetschichtspeicher.

**Hexadezimalsystem** oder auch Sedezimalsystem. Ein Zahlensystem mit der Basis 16 wird so bezeichnet. Die ersten 10 Ziffern dieses Systems bezeichnet man mit Null bis Neun, die folgenden 6 Ziffern mit den Buchstaben A – F (dezimal 10 – 15)

Hexadezimal	Binär	Dezimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

**KILO BYTE** oder KB genannt, steht für jeweils 1024 Bytes.



alphaTronic

## ZUM AUFWÄRMEN

### Laden eines Programmes

Die Konsequenzen der Kenntnis daraus: Ihr Computer verfügt über eine Anzahl von nicht flüchtigen Speicherstellen, in welchen sich das MOS-Monitor-Programm befindet. Dieses MOS-Monitor-Programm dient unter anderem dazu, in die flüchtigen Speicherstellen verschiedene Programme zu laden.

Wie lädt man nun ein Programm?

Dafür gibt es mehrere Möglichkeiten. Welche die richtige ist, hängt von verschiedenen Kriterien ab: Anwendungszweck, Systemkonfiguration, Programmebene.

Erste Möglichkeit: Eingabe eines Programms mit Hilfe eines MOS-Monitors über Tastatur, Speicherstelle für Speicherstelle.

Zweite Möglichkeit: Eingabe eines Basic-Compiler-Programmes über Tastatur. Solchermaßen einmal eingegebene Programme können in der Folge aus den flüchtigen Speicherbereichen auf andere Speichermedien übertragen werden, die nicht flüchtig sind. Zum Beispiel Cassette oder Diskette. So ist es möglich, ein in stundenlanger Arbeit erstelltes Programm in Sekundenschnelle von diesen externen Speichermedien Cassette oder Diskette in den flüchtigen Speicher zu laden und immer wieder zu verwenden, auch wenn zwischenzeitlich das Gerät abgeschaltet war.

Nächste Möglichkeit: Laden eines Programmes von Cassetten.

Nächste Möglichkeit: Laden eines Programmes von Diskette.

#### Laden eines Programmes von der Diskette

Wurde das Programm mit dem sog. Batchvorspann abgespeichert (alle Systemprogramme), so ist folgendes einzugeben:

- . B (innere SP-Taste drücken) Es wird File 0 anpositioniert, eingelesen und gestartet
- . B 1 (innere SP-Taste drücken). Es wird File 1 anpositioniert, eingelesen und gestartet.

Wurde das Programm nicht mit dem Batchvorspann abgespeichert, so ist folgende Kommandofolge erforderlich:

- . P | 1 (SP-Taste drücken). Es wird File 1 anpositioniert (S. 33)
- . l (SP-Taste Drücken). Es wird File 1 eingelesen und ab Adresse 4010 H abgelegt (S. 36).
- . G (Sp-Taste drücken). GO (Sprung) zur Start-Adresse 4010 H

**RAM.** Random Access Memory steht für Scheib-/Lesespeicher mit wahlfreiem Zugriff

**ROM.** Read Only Memory. Festwertspeicher mit wahlfreiem Zugriff, der vom Hersteller mit einem nicht mehr veränderbaren Inhalt geliefert wird.

**Speichermedien:** Außer diesen am einfachsten verfügbaren und relativ kostengünstigen externen Speichermedien gibt es noch andere Möglichkeiten: Hard-Disk, Magnetbänder, Magnetkarten, Lochstreifen, Lochkarten etc. und in Kürze auch Magnetblasenspeicher.

Kommandos sind bestimmte, vereinbarungsgemäß getroffene Anweisungen, die den Computer vorher festgelegte Programmsequenzen durchlaufen lassen. Verschiedene Programmiersprachen verfügen über unterschiedliche Befehlsvorräte. Je höher der Standard der Programmiersprache ist (höhere Programmiersprachen), desto einfacher ist die Befehlsstruktur für den Anwender.

Wir unterscheiden drei Arten von Anweisungen, bzw. Kommandos:

MOS (Mikrocomputer-Operating-System)-Kommandos sind Anweisungen an das Betriebssystem Ihres KISS. Diese Kommandos werden unmittelbar nach der Eingabe ausgeführt. Sie dienen u. a. zur Steuerung des E/A-Managements, zum Einlesen, Abspeichern und Durchführen von Programmen.

## **MOS-Kommandos**

Basic-Kommandos stellen eine Untermenge des Basic-Compilers dar, d. h. diese Kommandos sind Anweisungen an den Compiler und werden von diesem unmittelbar nach der Eingabe ausgeführt. Sie dienen zur Umwandlung Ihres Basic-Programmes in ein Maschinenprogramm, zum Testen, Abspeichern und Laden von Programmen usw.

## **BASIC-Kommandos**

Das sind jene Sprachelemente des Basic-Compilers, die dem Programmierer zur Verfügung stehen, um eine bestimmte Aufgabe mit Hilfe des Computers zu lösen.

## **Programmbefehle**



## **GRUNDLAGEN**

### **Was muß ich über Hardware wissen?**

Um es gleich vorweg zu nehmen:

Wenn Sie wissen, wie Sie Ihren Computer einschalten, dann wissen Sie bereits alles, was Sie zum Thema Hardware *unbedingt* wissen müssen.

Andererseits gibt es doch einige Gründe, über grundlegende Dinge etwas besser Bescheid zu wissen:

Erstens erleichtert Ihnen dieses Verständnis den Umgang mit Ihrem Gerät auch dann, wenn Ihre Neigungen in der Hauptsache auf der Software-Seite liegen. Zweitens ist es die Hardware, welche die Grenzen für Ihre Software-Möglichkeiten festlegt, man muß keineswegs komplexe Prozeß-Steuerungsaufgaben in Angriff nehmen, um den Nutzen zu erkennen, den Sie aus Ihrem Verständnis um die grundlegenden Vorgänge in Ihrem Computer ziehen. Spätestens wenn Sie Ihren Cassetten-Recorder anschließen, wird Ihnen Ihr Wissen zugute kommen.

Wir werden keineswegs versuchen, Sie zu überzeugen, Ihrem Computer mit dem Lötkolben zu Leibe zu rücken. Vielmehr sind es prinzipielle logische Verhaltensweisen, bzw. deren technische Realisation, auf die wir uns überall dort beschränken werden, wo wir aus Gründen des besseren Gesamtverständnisses auf das Thema Hardware eingehen. Für tiefergehende Information verweisen wir auf die speziellen technischen Dokumentationen, bzw. auf allgemeine Fachliteratur. Bitte beachten Sie die entsprechenden Hinweise in diesem Manual. Standard Fachliteratur finden Sie übrigens auch bei Ihrem Fachhändler, der Ihnen nach Möglichkeit bei der Beschaffung behilflich ist.



alphaTronic

## **GRUNDLAGEN** **Was muß ich über Software wissen?**

Es gibt einen Weg, den Umfang mit Software auf die reine Anwendung zu beschränken: man kann aus einem Angebot verschiedener Anwender-Software-Programme erwerben und diese am Computer dann einfach ablaufen lassen.

Das ist eine Methode der Beschränkungen, möglicherweise finden Sie in einer der ständig wachsenden Programmbibliotheken das eine oder andere Programm für Ihren Bedarf. Oft sind jedoch auch bei solcherart erworbenen Programmen Modifikationen notwendig, um sie exakt den eigenen Bedürfnissen anzupassen. Das ist ein Grund, über Software Bescheid zu wissen. Es gibt aber noch andere Gründe:

Natürgemäß beschränkt sich das Angebot fertiger Software auf häufig gefragte Programme. Für spezielle Wünsche ist die Auswahl stark eingeeengt.

Ein Grund, selbst zu programmieren!

In Ihrem System stecken Möglichkeiten, die es gilt auszunutzen, eine Palette von Systemsoftware macht das möglich.

Ein weiterer Grund, selbst zu programmieren!

Höhere Programmiersprachen machen es möglich, in wenigen Stunden Ihr eventuell erstes eigenes Programm laufen zu lassen, eine interessante und lehrreiche Beschäftigung. Sicherlich einer der befriedigendsten Gründe, sich mit Software zu beschäftigen.

Da es den Rahmen dieses Manuals sprengen würde, in allen Programmebenen allen Ansprüchen gerecht zu werden, finden Sie in dieser Broschüre zunächst einen allgemeinen Überblick über die verschiedenen System-Software-Möglichkeiten, sowie in der Folge jeweils einen Kurzaufsatz.

Entsprechend der verschiedenen Konfigurationen bzw. für die verschiedenen Programm-Sprachen stehen spezielle ausführliche Dokumentationen zur Verfügung. Beachten Sie bitte die Hinweise, oder fragen Sie Ihren Fachhändler.



Der MOS-Monitor (Micro-Computer-Operating-System) besteht aus einer Reihe von Programmen zur Abwicklung der MOS-Kommandos und des E/A - Managements.

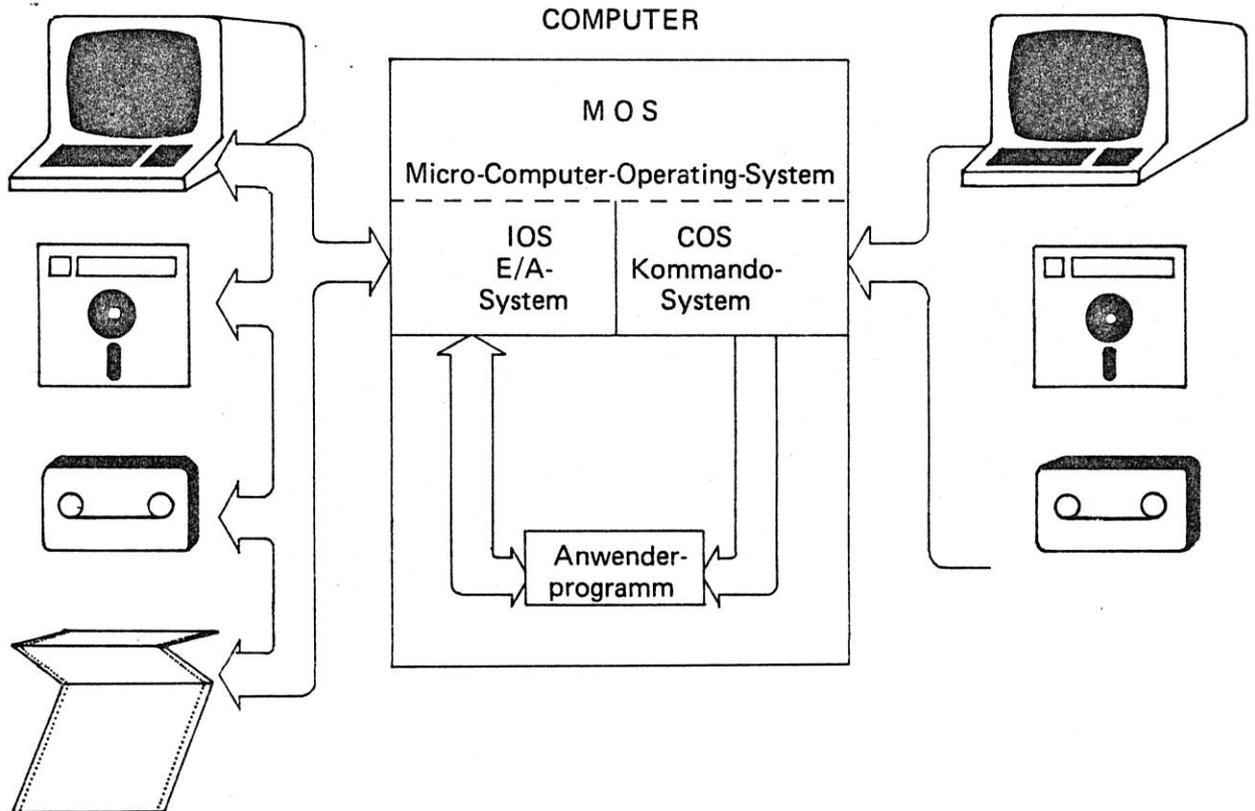
Der MOS-Monitor liest, analysiert und verarbeitet den JOB-Eingabe-Strom. Der JOB-Eingabestrom ist eine Folge von Anforderungen an Programmen sowie Dateien und Einheiten, welche von dieser verwendet werden.

#### Aufruf des MOS-Monitors

- Nach dem Betätigen des Netzschalters an Ihrem Gerät meldet sich automatisch der MOS-Monitor, da dieser in einem sog. EPROM-Speichermodul abgelegt und deshalb speicherresident ist.
- Die zweite Möglichkeit zum Aufrufen Ihres MOS-Monitors besteht durch das Drücken der RESET-Taste, siehe Tastenfunktionen.

Geräte-Ein-/Ausgabe

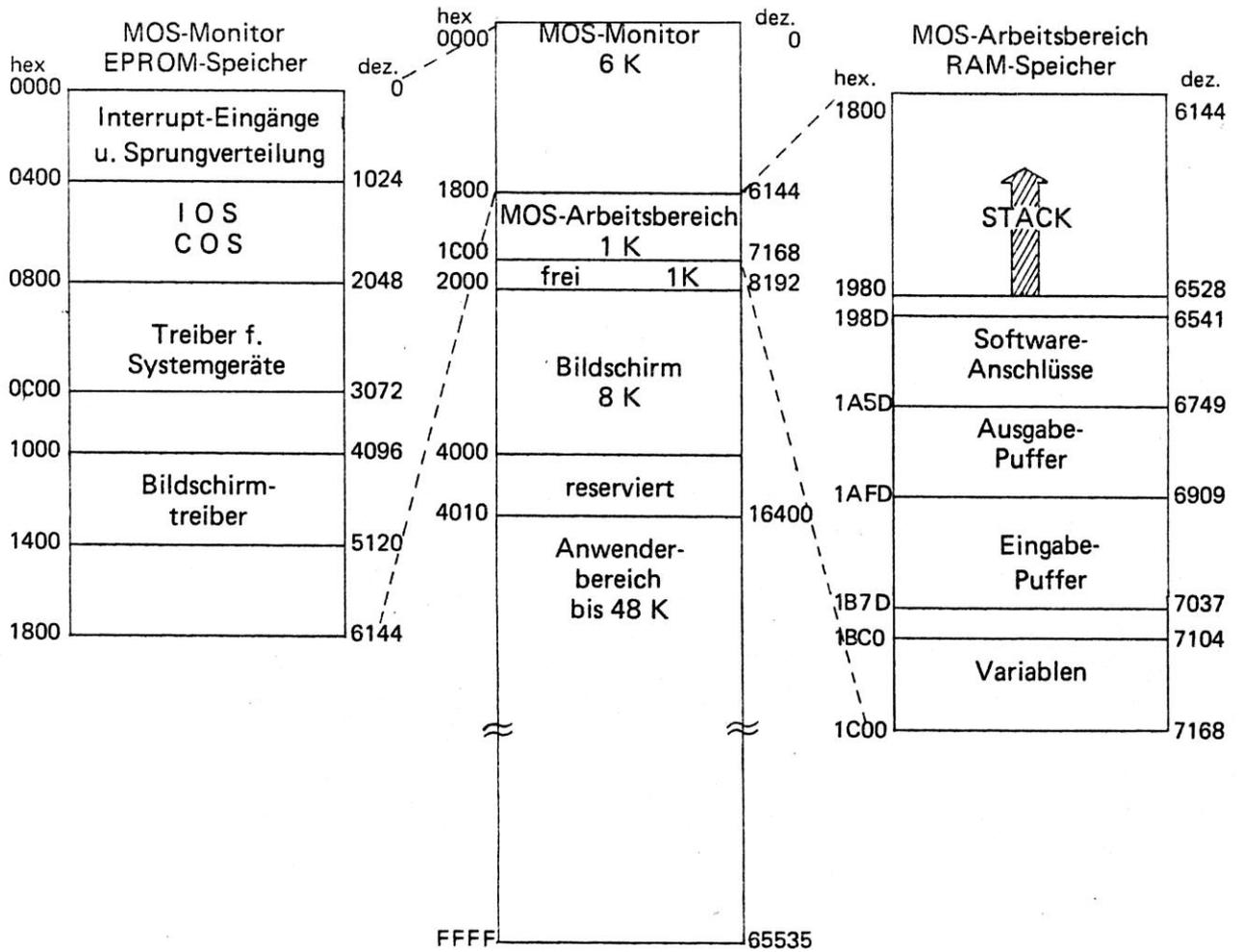
Steueranweisungen





alphaTronic

# MOS-MONITOR Speicher-Belegung





alphatronic

## MOS-MONITOR

### MOS-Kommandos

Das COS erhält seine Befehle in der Regel vom Benutzer durch Eingabe von Kommandos. Diese Kommandos bilden eine einfache Form von JOB-Control-Language. Da die Eingabe der Kommandos über die sogenannten Ströme des IOS geschieht, wird die Möglichkeit einer echten Stapelverarbeitung eröffnet, sh. Batch-Kommando.

Die Kommandos bestehen aus einem Kommandocharacter, dem u. U. Parameter folgen und werden mit der inneren Taste SP abgeschlossen. Sind mehrere Parameter anzugeben, so sind diese durch ein Komma voneinander zu trennen.

Manche Parameter besitzen eine sogenannte Voreinstellung, d. h. bei fehlender Angabe dieses Parameters wird ein bestimmter Wert angenommen.

#### Beispiel

0 xxxx, xxxx, ..... SP-Taste

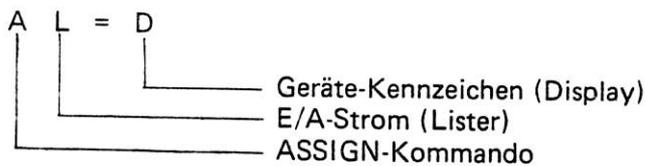
Adreßangaben können ohne führende Nullen eingegeben werden.

#### Syntax

Da das Mikrocomputer-Operating-System, kurz MOS genannt, über ein sog. geräteunabhängiges Datenmanagement-System verfügt, ist es erforderlich, dem MOS mitzuteilen, von bzw. zu welchem Gerät die einzelnen Datenströme gehen.

Wenn Sie z. B. eine für den Drucker vorgesehene Listausgabe nicht drucken, sondern auf dem Bildschirm sehen möchten, so wird dies durch folgendes Kommando bewirkt:

## ASSIGN



**ASSIGN** Zuordnen von Geräten und Strömen, d. h. der unter n angegebene E/A-Strom wird auf das unter g definierte Gerät gelenkt (Einstellen der Assignierschalter).

## .An-g

n = E/A-Strom, wobei  
C entspricht Command Input/Output  
R entspricht Reader  
W entspricht Writer  
L entspricht Lister

g = Geräte-Kennzeichen

Folgende Zuordnungen sind möglich:

C = C Tastatur/Display  
C = B Sys-in/Display (Batch)

R = S Sys-in  
R = K Tastatur

W = S Sys-Out  
W = D Display

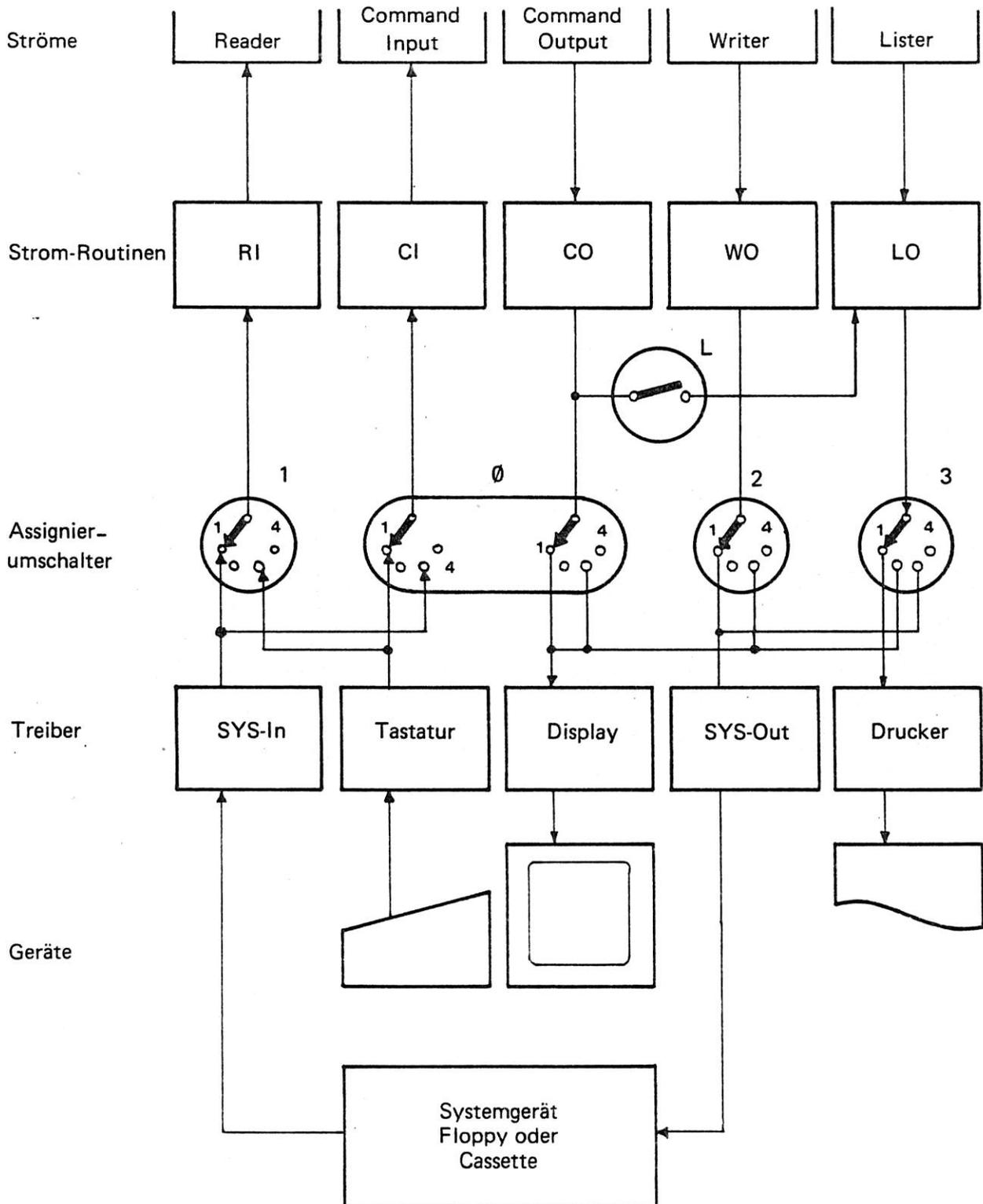
L = P Printer  
L = D Display  
L = S Sys-Out

Die Voreinstellung für die Zuordnung von Strömen und Geräten ist wie folgt vereinbart:

C = C, R = S, W = S, L = P

Zum besseren Verständnis soll Ihnen das nachfolgende Schaubild dienen.

## E/A- Struktur des IOS





Dieses Kommando bewirkt eine Einstellung bzw. Positionierung der Geräte, welche Sys-In oder Sys-Out zugeordnet sind, und muß stets dann gegeben werden, wenn Sie über MOS- oder BASIC-Kommandos eine Eingabe oder Ausgabe veranlassen möchten.

## **PRESET DEVICE**

.P = MOS-Kommando für PRESET DEVICE

**.Pgpl,p2,p3**

g = Angabe des entsprechenden Gerätes, für das die Einstellung vorgenommen werden soll.  
I entspricht Sys-In (Floppy oder Cassette)  
O entspricht Sys-Out (Floppy oder Cassette)  
P entspricht Drucker.

p 1 = Parameter 1:

- a) Hier ist die File-Nr. bei Sys-In oder Sys-Out anzugeben, da sich auf einer Diskette auch mehrere Files, bzw. Dateien befinden könnten.
- b) Beim Drucker können hier 2 Bytes zur Einstellung einer anderen Druckart angegeben werden. Siehe MOS-Manual. Ø = keine Änderung der Einstellung.

p 2 = Parameter 2:

- a) Bei mehrspurigen Cassettenlaufwerken, z. B. DC 100, kann hier die jeweilige Spur eingestellt werden. Bei Disketten ist stets Ø anzugeben.
- b) Beim Drucker können hier 2 Bytes zur Einstellung der V24-Schnittstelle angegeben werden.
  - 1. Byte = Mode für USART 8251 von Intel z. B. 4EH entspricht asynchrone Übertragung von 8 Bit mit 1 Stopbit und No parity
  - 2. Byte = Einstellung der Baudrate, siehe Tabelle, Seite 34

p 3 = Parameter 3:

- a) Bei Sys-In oder Sys-Out kann hier die Nummer des entsprechenden Diskettenlaufwerkes angegeben werden. Ø entspricht dem 1. Laufwerk, 1 entspricht dem 2. Laufwerk usw.
- b) Beim Drucker ist dieser Parameter nicht belegt.



Voreinstellung:

- . P 0 0,0,0                    Sys-Out
- . P 1 0,0,0                    Sys-In
- |                          p 3,            Laufwerks-Nr. 0
- |                          p 2,            Spur-Nr. 0
- |                          p 1,            File-Nr. 0
- . P P 1 E 0 5,4 E 0 C, 0    Drucker

**PRESET DEVICE**

Beispiele:

- . P 1 0,0,0    =    Einstellen Sys-In auf Laufwerk 0, File 0
- . P 1            =    wie oben, Kurzform
- . P 0 2         =    Einstellen Sys-Out auf Laufwerk 0, File 2
- . P 0 3,,1     =    Einstellen Sys-Out auf Laufwerk 1, File 3

Parametertabelle zur Einstellung der Baudrate:

Baudrate	Parameter
9600	06
7200	08
4800	0C
3600	10
2400	18
1800	20
1200	30
900	40
300	C0



Mit diesem Kommando wird auf das eröffnete Writer-Gerät (Sys-Out oder Display) der Inhalt des RAM-Speichers ausgegeben.

Die Zuordnung des Writer-Gerätes erfolgt mit dem Kommando

- . A W = S für Sys-Out und
- . A W = D für Display, siehe ASSIGN

Die Eröffnung des Writer-Gerätes erfolgt mit dem PRESET DEVICE-Kommando . P 0 p1, p2, p3

Der Inhalt des Speichers wird von Adresse a bis Adresse e in der Blockgröße b ausgegeben.

- a = Anfangsadresse, hexa-dez.
- e = Endadresse, hexa-dez.
- b = Blockgröße, hexa-dez., max. 2 K

Beispiel

- |                     |   |
|---------------------|---|
| . AW = S            | Sys-Out dem WriterGerät zuordnen.   |
| . P01               | Sys-Out eröffnen und auf File 1 einstellen (File 0 muß bereits angelegt sein)                     |
| . 04010, 5FF0, 07EF | Ausgeben des Speichers von Adresse 4010H bis 5FF0H mit der Blockung 7EFH auf die Diskette, File 1 |
| . AW = D            | Sys-Out dem Display zuordnen  |
| . P 0               | Sys-Out eröffnen  |
| . 04010, 5FF0, 7EF  | Ausgeben des Speicherinhaltes auf dem Bildschirm  |

Voreinstellung:

- a = 4010 H
- e = 5FF0 H
- b = 07EFH

OUTPUT ist nur auf das eröffnete und eingestellte Writer-Gerät möglich, d. h. ein PRESET DEVICE-Kommando für das Gerät muß vorausgegangen sein.

Da automatisch keine Filemarke ausgegeben wird, erscheint die Meldung EOF?

Die Filemarke muß durch das Endfile-Kommando ausgegeben werden.

**OUTPUT**

**.a,e,b**



alphatronic

## MOS-MONITOR

MOS-Kommandos

Der Writer-Strom wird mit einer File-Marke abgeschlossen. Dies ist stets nach einer Ausgabe auf Sys-Out erforderlich, also bei Diskette oder Cassette, um das Datei-Ende zu kennzeichnen.

**ENDFILE**

**.E**

Vom eröffneten Reader-Gerät wird ein File bis zur Filemarke oder der Endadresse eingelesen und ab Anfangsadresse abgespeichert.

**INPUT**

**.a,e**

a = Anfangsadresse, hexa-dez.  
e = Endadresse, hexa-dez.

Das Ablegen erfolgt Byte für Byte ab Fileposition, d. h. der File enthält keinen Vorspann und keine Ladeadressangaben (Batch-Vorspann, siehe Batch-Mode, bzw. Hex-Kommando).

INPUT ist nur vom eröffneten und eingestellten Reader-Gerät möglich, siehe OUTPUT.

Beispiel

. AR = S	Sys-In wird dem Reader-Gerät zugeordnet.
. P I 1	Sys-In eröffnen und auf File 1 einstellen.
. I6100, FFEF	Einlesen des Files und Ablegen der Daten von Adresse 6100H bis FFEFH, bzw. bis die Endfilemarke gelesen wurde.

Voreinstellung:

a = 4010H  
e = FFEFH



# MOS-MONITOR

MOS-Kommandos

Dies ist eine Sprunganweisung zu der unter a angegebenen Adresse. Das heißt, der Programmcounter wird auf die Adresse a eingestellt

a = Start-Adresse, hexa-dez.

Beispiel

- . G4010 bewirkt einen Sprung zur Adresse 4010 H. Falls Sie vorher den Basic-Compiler geladen haben, wird dieser gestartet.
- . G6100 Ein Programm auf Adresse 6100 H wird gestartet.
- . G55 bewirkt einen sog. Warmstart des MOS-Monitors.

Für Assembler-Programmierer:

Das G0-Kommando bewirkt das Laden der Register in der Reihenfolge PSW, RB/RC, RD/RE, RH/RL aus dem augenblicklichen Stack und dem Programmcounter mit der unter a angegebenen Adresse. Fehlt dieser Wert, so wird er ebenfalls aus dem Stack geholt, was zugleich dem Hex-Wert aus der MOS-Grundstellungsanzeige entspricht.

RESET

KISS-MOS

\$1976 4010 FFEF



Ersetzen des Inhalts der unter a angegebenen Adresse, d. h. der Inhalt wird angezeigt und kann nun verändert werden.

Nach Eingabe des neuen Inhalts mit Blank als Abschluß wird das nächste Byte angezeigt. Bei Eingabe von nur Blank bleibt der alte Wert erhalten und das nächste Byte wird angezeigt, bei - wird ein Byte zurückgegangen.

Beendet wird SUBSTITUT mit der SP-Taste.

**G0**

**.G2**

**SUBSTITUTE**  
**.Sa**



alphatronic

## MOS-MONITOR

MOS-Kommandos

Der Protokoll-Schalter wird sozusagen umgelegt, d. h. aus EIN wird AUS und umgekehrt. Bei EIN wird jede Ausgabe des Command-Outputs auch auf den Lister geführt, wobei zuvor jedoch der Lister mit dem PRE-SET DEVICE-Kommando eröffnet werden muß.

Beispiel

.PP

.L

Nach dem Einschalten bzw. RESET steht der Protokoll-Schalter auf AUS.

### LIST COMMAND OUTPUT

.L

Der Speicherbereich von a bis e wird nach z umgespeichert.

a = Anfangsadresse, hexa-dez.

e = Endadresse, hexa-dez.

z = Zieladresse, hexa-dez.

### MOVE MEMORY

.M a,e,z

Beispiel

.M6100,7000,9000

Der Speicherbereich von a bis e wird über den Lister hexa-dezimal ausgegeben.

a = Anfangsadresse, hexa-dez.

e = Endadresse, hexa-dez.

### DISPLAY MEMORY

.Da,e

Beispiel

.D6100,7000

Bitte beachten Sie vor der Anwendung dieses Kommandos, daß dem Lister sowohl der Bildschirm (Display), als auch der Printer oder Sys-Out zugeordnet werden kann.



## MOS-MONITOR

MOS-Kommandos

Der Speicherbereich von a bis e wird mit dem unter w angegebenen Wert gefüllt.

a = Anfangsadresse, hexa-dez.  
e = Endadresse, hexa-dez.  
w = Füllwert, hexa-dez.

### FILL MEMORY

**.Fa,e,w**

Der Speicherbereich von a bis e wird byteweise ab der Vergleichsadresse v auf Übereinstimmung geprüft. Bei Ungleichheit wird die Byte-Adresse angezeigt.

a = Anfangsadresse, hexa-dez.  
e = Endadresse, hexa-dez.  
v = Vergleichsadresse, hexa-dez.

### COMPARE MEMORY

**.Ca,e,v**

Mit den beiden Kommandos FILL MEMORY und COMPARE MEMORY läßt sich sehr einfach ein Speichertest durchführen.

Beispiel

- . F 4000, FFFF, FF Speicher von Adresse 4000H bis FFFFH mit FFH füllen
- . C 4000, 9FFF, A000 Speicher von Adresse 4000H bis 9 FFFH mit Speicher von Adresse A 000H bis FFFFH vergleichen.

Einstellen und Anzeigen der höchsten Speicheradresse. Wird m nicht angegeben, so wird die eingestellte Adresse angezeigt.

m = Speicheradresse, hexa-dez.

Voreinstellung: FFEFH

### TOP OF MEMORY

**.Tm**

Eine benutzereigene Routine mit gegebener absoluter Eingangsadresse wird an einen sogenannten Softwareanschluß des MOS angehängt.

a = Anschluß-Nr., hexa-dez., siehe MOS-Manual  
e = Eingangsadresse, hexa-dez.

### EXTEND MOS

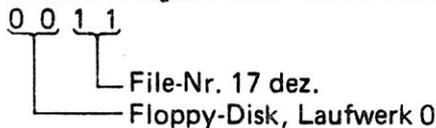
**.Xa,e**



**BATCH MODE**  
**.Bsf,io**

Es wird ein internes Preset Device Kommando mit Spur und File auf Laufwerk 0 durchgeführt, wobei unter Spur das Floppy-Laufwerk zu verstehen ist. Anschließend wird die weitere Kommandoeingabe über Sys-in erwartet.

sf = 4stellige hexadezimale Angabe über Gerät und File-Nr., z. B. 0 0 1 1



io = IO-Byte, siehe Seite 41

Hiermit kann das IO-Byte durch Angabe eines 4stelligen Hex-Wertes verändert werden. Das IO-Byte wird mit den ersten beiden Zeichen durch log. UND verknüpft und mit den letzten beiden Zeichen mit log. ODER.

io = FF00, d. h. es erfolgt keine Veränderung der Zuordnung von Geräten und Strömen.

io = 0040, d. h. das gesamte IO-Byte wird gelöscht (UND) und mit den beiden letzten Zeichen gefüllt (ODER).

IO-Byte	10100010	alter Zustand
AND	00000000	Zeichen 1+2 (00)
Ergebnis	00000000	
ODER	01000000	Zeichen 3+4 (40)
Ergebnis	01000000	neuer Zustand

**USER PROGRAMM**  
**.Us,io**

Das io-Byte im MOS (siehe S. 41) wird mit dem oberen Byte der io-Maske mit log. UND verknüpft und mit dem unteren Byte mit log. ODER.

Anschließend wird direkt bei der unter s angegebenen Startadresse fortgefahren.

s = Startadresse, hexa-dez.

io = io-Byte-Maske.



alphaTronic

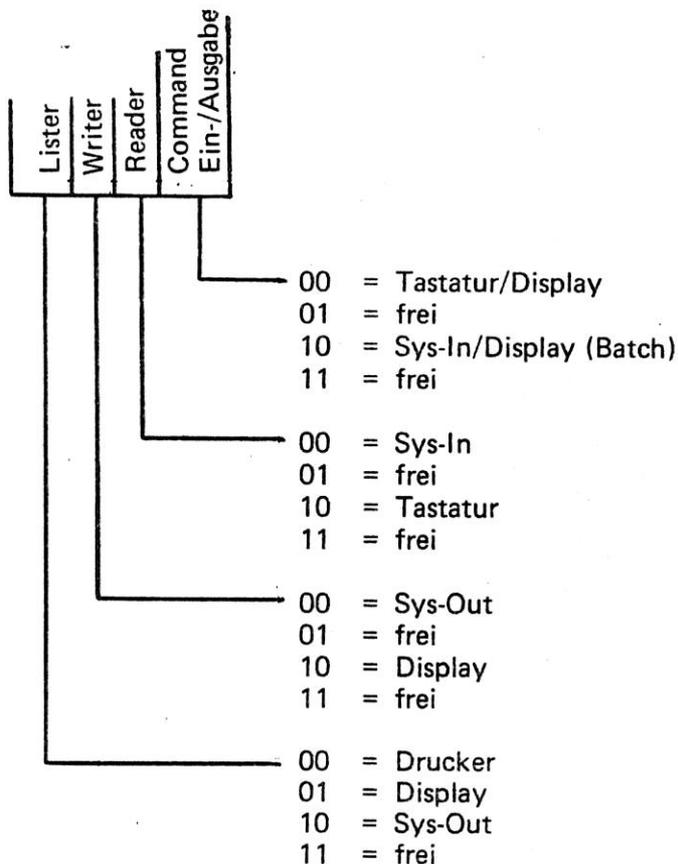
# MOS-MONITOR

## MOS-Kommandos

Das IO-Byte repräsentiert die jeweilige Zuordnung der E/A-Ströme zu den Geräten. Eine Veränderung dieser Zuordnung wird in der Regel durch das ASSIGN-Kommando vorgenommen. Ausnahmen bilden die Kommandos BATCH MODE und USER PROGRAM. Per Programm kann die Zuordnung der Geräte dynamisch zur Ausführungszeit mit dem Parameter DEV (d) verändert werden.

### IO - Byte

- Die Veränderung des IO-Bytes erfolgt stets in 2 Stufen:
1. Bestehende Einstellung löschen (mittels UND-Verknüpfung)
  2. Neue Einstellung setzen (mittels ODER-Verknüpfung)





alphaTronic

# BASIC COMPILER

Das Arbeiten und Experimentieren mit Microcomputern erfordert neben einigen wichtigen Kenntnissen in der Hardware in erster Linie fundierte Programmierkenntnisse.

Dieses Kapitel soll Ihnen bei der Einarbeitung in dieses heute so wichtige Gebiet helfen. Von der Grundlage der Programmierung ausgehend, werden die Handhabung des Basic-Compilers und die einzelnen Basic-Befehle kurz beschrieben. Eine tiefergehende Beschreibung des Basic Compilers finden Sie im Basic Manual.

## Allgemeines

Basic ist diejenige Computersprache, die heute von vielen Elektronikern und Fachleuten beherrscht wird. Sie gehört an vielen Schulen zum Lehrstoff und ist auch für den Laien leicht erlernbar.

Dadurch, daß viele Leute diese Computersprache beherrschen, gibt es auch viele Programmierbeispiele in dieser Sprache. Ganze Programmbibliotheken für die verschiedenen Anwendungsbereiche, wie Buchhaltungsprogramme für Kleinbetriebe, mathematische Funktionen, Computerspiele u. v. a. stehen heute dem Computerhobbyisten zur Verfügung.

BASIC ist die Abkürzung für „Beginners All-Purpose Symbolic Instruction Code“ (Programmiersprache für Anfänger und universellen Einsatzbereich). Sie wurde vor einigen Jahren am Dartmouth College in den USA entwickelt.

Für den Computer steht Ihnen BASIC/C zur Verfügung. Das BASIC-C-System ist ein Programm-Paket, welches dem Benutzer die Lösung von Problemen und Aufgaben aus dem kommerziellen und technisch-wissenschaftlichen Bereich erlaubt.

## Einführung



## Basic Compiler

### Speicherbelegung und Arbeitsweise

Der BASIC-Compiler steht Ihnen auf Diskette zur Verfügung. Um also mit dem Compiler arbeiten zu können, ist es erforderlich, diesen in den Speicher zu laden.

### Laden des Compilers

1. Gerät einschalten
  - Das MOS meldet sich (Seite 17) und ist bereit zur Aufnahme von Kommandos.
2. Systemdiskette einlegen
  - Seite 18
3. B eingeben und die innere SP-Taste drücken
  - Dieses sog. Batch-Kommando veranlaßt das MOS, den Basic-Compiler aus File 0 der Diskette in den Speicher zu laden. Der Bildschirm wird gelöscht, und es erscheint folgende Meldung:

```
. I 4010 5FFF
? I B000 FFE0
#0200
. G
  BASIC V (2.04)

*** NEW oder OOLD? ***
```

Der Basic-Compiler möchte nun von Ihnen wissen, ob Sie sein neues Programm erstellen möchten oder ein altes, d. h. ein bereits bestehendes Programm ändern oder ergänzen wollen. Da Sie alle Kommandos auch in der Kurzform eingeben können, ist es auch zur Beantwortung dieser Frage nur erforderlich, N oder 0 einzugeben.

Bei N löscht der Compiler den Speicher, bringt das System in Grundstellung und meldet sich mit READY. Bei 0 Wird der Speicher *nicht* gelöscht und bereits gespeicherte Programme können ergänzt oder verändert werden.

```
BASIC V (2.04)
***NEW OR OLD?***
N
READY!
```

```
BASIC V (2.04)
***NEW OR OLD?***
0
READY!
```



# Basic Compiler

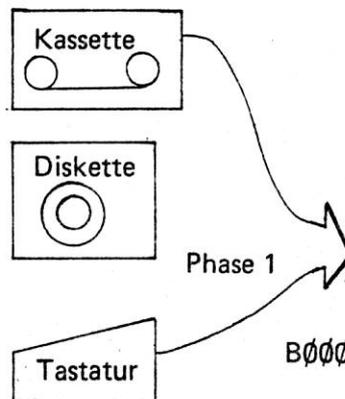
## Speicherbelegung und Arbeitsweise

### Phase 1

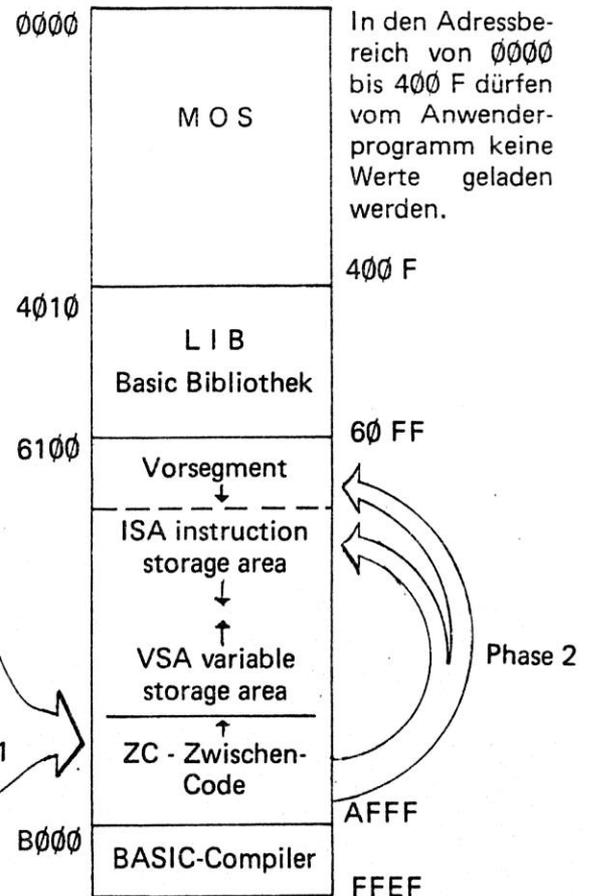
Das Basic-Quellenprogramm kann über die Tastatur, das Cassettengerät oder über Diskette eingelesen werden.

Dieses Quellenprogramm wird in einen sog. „Zwischencode“ umgewandelt und in komprimierter Form im Speicher abgelegt.

Mit den verschiedenen Basic-Kommandos kann dieser Zwischen-Code (ZC) angezeigt, gelistet, geändert, ergänzt u. a. werden.



Speicherbelegungsliste (hexa)

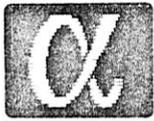


### Phase 2

In dieser Phase wird der Zwischen-Code in den Maschinencode umgewandelt. Zugleich findet eine Code-Optimierung und eine semantische Prüfung statt.

Der generierte Maschinencode wird in der sog. ISA (instruction storage area) ohne Variablen und Konstanten abgelegt. Der Bereich der Variablen (VSA) wird bei der Compilierung berechnet, jedoch nicht belegt, da dieser vom Basic-Programm erst zur Ausführungszeit benötigt wird.

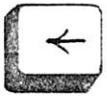
Alle Konstanten des Programms werden im Vorsegment der ISA abgelegt. Innerhalb dieses Vorsegmentes befinden sich auch die Vorbelegungsroutine zum Löschen der Variablen, sowie Hilfsfunktionen zum Starten des Programms und eine Tabelle der verwendeten Ein-/Ausgabe-Routinen.



alphaTronic

# BASIC-Compiler

## Zeichen/Sonderzeichen



oder

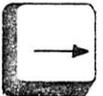


Drücken dieser Tasten bewirkt das Löschen vorangehender Zeichen, jedoch erst, wenn Sie durch Betätigung der inneren SP-Taste die Zeile an den Basic-Compiler abgeben.

### Korrigieren bei Eingabefehlern



Löschen der eingegebenen Zeile



Übernahme des an der Cursor-Position stehenden Zeichens



Abbruch der Ausgabe beim nächsten Zeilenwechsel.

### Abbruch von Ausgaben



Bewirkt bei LIST oder SAVE, siehe Basic-Kommandos, den Abbruch der Ausgabe beim nächsten Zeilenwechsel.

Der Zeichenvorrat des Basic-Compilers umfaßt alle 26 Zeichen des Alphabets, die 10 Ziffern und folgende Sonderzeichen:

### Zeichenvorrat

␣ Zwischenraum	- Minus
+ Plus	/ Schrägstrich
* Stern	( linke Klammer
, Komma	) rechte Klammer
\$ Dollar	. Punkt
< kleiner	≠ ungleich
" Doppelapostroph	> größer
= Gleichheitszeichen	% Prozent

Alle anderen druckbaren Zeichen sind nur in Kommentaren und Strings (Zeichenketten) zulässig. Die o. a. Sonderzeichen zum Berichtigen von Eingaben und zum Abbruch von Ausgaben gehören nicht zum Zeichenvorrat des BASIC-Compilers und dürfen daher weder in Kommentaren noch in Strings verwendet werden.



Wie wir wissen, ist Basic eine interaktive Sprache, d. h. Programme werden im Dialog zwischen dem Benutzer und dem Computer ein- und ausgegeben, verändert und zur Ausführung mittels sog. *Basic-Kommandos* gebracht. Sie bestehen in der Regel aus einem Kommandowort, das von einigen Parametern gefolgt sein und vom System sofort ausgeführt werden kann.

Das Kommando @ LIST 50, 100, 1 bewirkt, z. B., daß alle Anweisungen von Nr. 50 bis Nr. 100 des im Speicher befindlichen Programmes auf dem List-Gerät nach dem Drücken der inneren „SP“-Taste protokolliert werden.

## Allgemeines

Alle Basic-Kommandos beginnen stets mit dem Zeichen „@“ und können von mehreren numerischen Parametern begleitet sein. Diese können entweder dezimal (z. B. 255) oder hexadezimal mit abschließendem H (z. B. FFH) eingegeben werden. Hexadezimale Werte können mit oder ohne führende Nullen eingegeben werden.

Alle Kommandoworte können bis zur Eindeutigkeit abgekürzt werden.

## Syntax

Für einige Basic-Commandos ist die vorherige Eingabe von MOS-Kommandos erforderlich. Dies kann auch unter der Steuerung des Basic-Compilers erfolgen. Zum Beispiel

@ @ P03 bewirkt ein Preset Output vor File 3  
@ @ PI6 Positionieren vor File 6

Der Eingabe von MOS-Kommandos wird also stets das @ -Zeichen zweimal vorangestellt. Nach Ausführung der MOS-Anweisung gelangt die Programm-Kontrolle wieder zum Basic-System, d. h. Sie können Basic- und MOS-Kommandos entsprechend Ihren Erfordernissen mischen.

z.B.:	Nr.	Kommando	Bemerkung
	1	@ NEW	Rücksetzen des Systems
	2	@ @ PI6	Positionieren vor File 6
	3	@ LOAD	Quellen-Progr. (File 6) einlesen
	4	@ LINK 6100H, 7FFFH, 5, 4	
	5	@ NEW	
	6	@ BEGIN 6400H	Ablageadr. setzen
	7	@ @ PI4	Positionieren vor File 4

## MOS-Kommandos



alphatronic

# BASIC COMPILER

## Basic-Kommandos

### Kommandos zur Texthaltung

Laden eines Quellen-Programms von Diskette oder Cassette von Anweisungsnummer i bis m bei gleichzeitiger lexikalischer Analyse, wobei fehlerhafte Anweisungen nicht übernommen werden. Bei entsprechender Numerierung des im Speicher befindlichen Programmes kann das neu zu ladende Quellenprogramm gezielt eingemischt werden.  
n = 1 d. h. gleichzeitige Anzeige auf dem Bildschirm.

@ **LOAD i,m,n**

@ **LOi,m,n**

Beispiel

@ @ P I 1        PRESET DEVICE, Sys-In, File 1

@ LOAD „1

Anmerkung: Vor dem LOAD-Kommando ist stets ein PRESET DEVICE zu geben.

---

Ausgabe eines Quellen-Programms von Anweisung-Nr. i bis m auf Diskette oder Cassette  
n 2 = 0 d. h. mit Datei-Abschluß (E O F)  
n = 1, d. h. ohne Datei-Abschluß

@ **SAVE i,m,n**

@ **SAi,m,n**

Beispiel

@ @ P O 2        PRESET DEVICE, Sys-Out, File 2

@ SAVE 10, 3000, 0

Anmerkung: Vor dem SAVE-Kommando ist stets ein PRESET DEVICE zu geben.

---

Speicher löschen und System in Grundstellung bringen

@ **NEW**

@ **N**

Eingang ins Basic-System, d. h. ein schon vorhandenes Programm kann weiter ergänzt werden.

@ **OLD**

@ **0**



# BASIC COMPILER

## Basic-Kommandos

### Kommandos zur Texthaltung

Ausgeben des Quellen-Programms von Anweisung-Nr. i bis m auf den Bildschirm. Ist n = 1, so wird gleichzeitig auf dem Drucker protokolliert.

@ **LIST i,m,n**

@ **LISi,m,n**

@ LIST Das gesamte Quellenprogramm wird angezeigt.

Anmerkung: Ist n = 1, so ist vorher ein PRESET-DEVICE zu geben (@@ PP)

Löschen der Anweisung eines Quellen-Programms von i bis m.

@ **DELETE i,m**

@ **DE i,m**

ACHTUNG: Werden die Parameter i und m nicht angegeben, wird das gesamte Quellen-Programm gelöscht.

@ DE 20 Zeile 20 wird gelöscht

@DE25,100 Zeile 25 bis einschließlich 100 wird gelöscht.

Anzeigen der Werte von Variablen, deren Basisadresse mit i, bzw. m angegeben wurden (nur Gleitkommazahlen)

@ **DUMP i,m**

@ **DU i,m**

Neunumerierung eines Quellen-Programms mit den Anweisungs-Nr. i bis m mit Schrittweite n und neuer Anfangs-Nr. o.

@ **RENUMBER i,m,n,o**

@ **RE i,m,n,o**

Wird n und o nicht angegeben, so wird n = 10 und o = 10 gesetzt.

Nach der Neunumerierung werden die niedrigste und die höchste Nummer angezeigt.



# BASIC COMPILER

## Basic-Kommandos

Einstellung der Start Adresse zur Ablage des Objekt-Codes.

Dieses Kommando ist stets von der COMPILE-Anweisung zu geben. Wird die BEGIN-Anweisung nicht gegeben, so wird als Start-Adresse stets 6100 H angenommen.

Beispiel

```
. BEGIN 6300H
. B 25344
```

Beide Kommandos stellen die Start-Adresse auf 6300H, bzw. 25344 dezimal.

### Kommandos zur Compilierung

```
@ BEGIN i
@ B
```

Erzeugen eines Objekt-Codes von Anweisungs-Nr. i bis m,

```
@ COMPILE i,m,n
@ CO i,m,n
```

	Code Erzeugung	Cross referenz	TRACE Vorbereitung	Symbol tabelle
n = 0	X			
n = 1	X	X		
n = 2	X		X	
n = 3	X	X	X	
n = 4	X			X
n = 5	X	X		X
n = 6	X		X	X
n = 7	X	X	X	X

Voreinstellung:

i = kleinste Anweisungs-Nr.  
m = größte Anweisungs-Nr.  
n = 0

```
@ NEW
10 CHAR A$ (1)
20 Let B = 8.5 E + 01
30 Let A$ = CHG (B)
40 PRINT A$
```

```
@ COM ,, 5
```

```
B ..... 6EFDH
A$ ..... 6EFAH
10 = 6120 H
20 = 6120 H
30 = 6129 H
40 = 6135 H
```

```
ISA 6100 H 613DH
VSA 6EFAH 6EFFH
```

```
@ RUN
U Ergebnis
```



alphaTronic

# BASIC COMPILER

Basic-Kommandos

Teststart eines mit COMPILE übersetzten Programmes von Anweisungs-Nr. i bis m.

n = 0, d. h. kein TRACE-Ausdruck

n = 1, d. h. mit TRACE-Ausdruck, sofern in der COMPILE-Anweisung die TRACE-Vorbereitung mit angegeben wurde.

Voreinstellung

i = kleinste Anweisungs-Nr.

m = größte Anweisungs-Nr.

n = 0

## Kommandos zur Compilierung

@ **RUN i,m,n**

@ **RUI,m,n**

Übersetzen des eingelesenen Quellen-Programmes in den Maschinencode mit anschließender Ausführung des erzeugten Objektprogrammes.

@ **START**

@ **ST**

Beispiel

@ N Grundstellung  
@@ PI PRESET DEVICE, File 0  
@ L 0 LOAD File 0 (Quellenprogramm)  
@ ST START (Assemblierung und Ausführung)

Abspeichern des Objektprogrammes auf Cassette oder Diskette.

i = 0, d. h. abspeichern mit Batch-Vorspann

i = 1, d. h. abspeichern mit Batch-Vorspann und Basic-Bibliothek

i = 2, d. h. abspeichern mit Basic-Bibliothek

Anmerkung: Ein mit Batch-Vorspann ausgelagertes Programm kann unter MOS mit dem sog. BATCH-Kommando (.Bi) anpositioniert, geladen und gestartet werden.

Wird die Basic-Bibliothek mit ausgelagert, so werden alle Teilprogramme zwischen der Bibliothek und dem letzten übersetzten BASIC-Programm mit abgespeichert.

@<sup>k</sup> **HEX i**

@ **HI**

Beispiel

@@ P 0 1 PRESET DEVICE, File 1  
@ H 1 Abspeichern des Objektprogrammes in File 1



# BASIC COMPILER

## Basic-Kommandos

### Kommandos zur Compilierung

Übersetzen eines Quellenprogrammes in den Maschinencode (Compile-Lauf).

@ **LINK i,m,n,o**

Im Gegensatz zur BEGIN/COMPILE-Anweisung besteht beim LINK-Kommando die Möglichkeit, die Anfangsadresse des Variablenbereiches (VSA) zu bestimmen.

Da der Bereich der Variablen (VSA, siehe S. 44) bei der Compilierung nur berechnet und nicht belegt wird, kann die VSA auch den Speicherplatz des Basic-Compilers oder des Zwischen-Codes belegen. Dieses Kommando wird man also immer dann anwenden, wenn der normalerweise verfügbare Speicher zwischen ZC (Zwischen-Code) und Basic-Bibliothek nicht ausreicht.

i: Anfangsadresse des zu erzeugenden Objektcodes

m: Anfangsadresse des variablen Bereiches

n: entspricht n bei COMPILE

o: Adresse, die den Anfangswert des Stackpointers beinhaltet. Der Standardwert befindet sich im MOS als Konstante und kann über den Wert 4 benutzt werden (o = 4)

### Systemdienste

Berechnen der Summe und Differenz von i u. m. Wird m nicht eingegeben, so wird i bei dezimaler Eingabe in eine hexadezimale Zahl umgewandelt und umgekehrt.

@ **CALCULATE i,m**

@ **CAi,m**

@ C A 32 Ergebnis: 0020H

@ C A 20 H Ergebnis: 32

Anzeige der Anfangs- und Endadresse des sich im Speicher befindlichen Objekt-Programmes.

@ **INF**

@ **I**

Anzeige des Zwischencodes der letzten eingegebenen Anweisungszeile

@ **Q**

Unterbrechen des Basic-Systems und Rücksprung ins MOS-System.

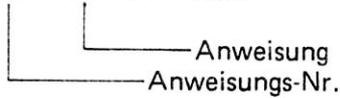
@ **X**

Fortsetzung an der Unterbrechungsstelle im MOS mit dem MOS-Kommando (.G)

### Syntax

BASIC-Programme bestehen aus einer Folge von Anweisungen, wobei jede Anweisung die Anweisungs-Nr. und die eigentlichen Anweisungen bzw. die Programmbefehle, enthält.

Z. B. 30 LET A = 3.75

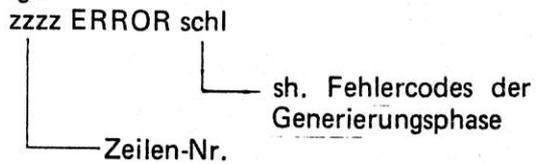


Anweisungen enthalten stets zwei Komponenten, d. h. ein Schlüsselwort, das die Art der Anweisung spezifiziert und eine Folge von Argumenten, die zur Ausführung der Anweisung erforderlich sind.

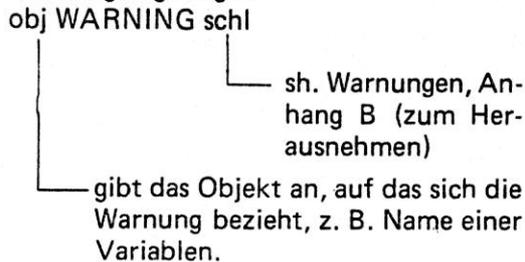
Entsprechen die Anweisungen nicht den für BASIC gültigen Regeln, so werden syntaktische Fehler unmittelbar nach Eingabe einer Zeile vom System erkannt und unterhalb der fehlerhaften Zeile durch ein ? markiert.

Der Markierung des Fehlercodes folgt in der nächsten Zeile ein Schlüssel, der Aufschluß über die Art des Fehlers gibt (sh. Fehlercodes der Analysephase). Fehlerhafte Zeichen werden nicht abgespeichert, sondern sind neu einzugeben.

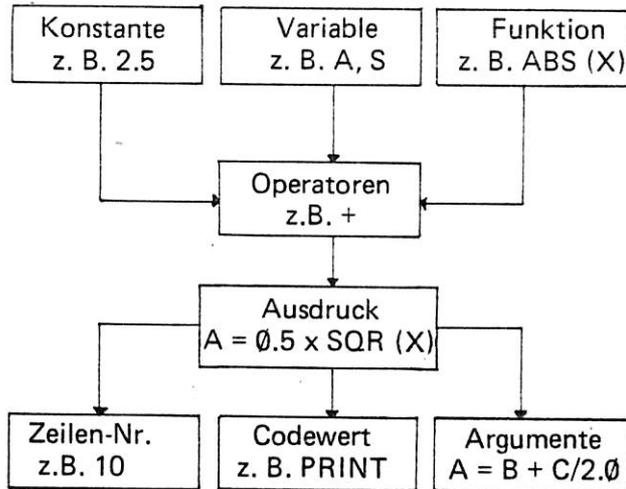
Semantische Fehler (z. B. unerlaubte Schleifenschachtelung) werden während der Compilierung in folgender Form angezeigt:



Liegt Verdacht auf einen Benutzerfehler vor, wobei jedoch keine grammatischen Regeln verletzt wurden, so wird dies als Warnung angezeigt.



## Sprachumfang



Auf den nachfolgenden Seiten ist der gesamte Sprachumfang des Basic-Compilers erläutert und kurz beschrieben. Eine weitergehende und ausführliche Beschreibung finden Sie im BASIC-COMPILER-MANUAL.



alphatronic

können bis zu 6 Buchstaben oder Ziffern lang sein, wobei jedoch das erste Zeichen ein Buchstabe sein muß. Das letzte Zeichen kann zur Typerkennung verwendet werden (\$ oder %)

- Als Variablen können nicht verwendet werden:
- Schlüsselwörter, wie z. B. IF, THEN, ELSE usw.
  - Operator-Namen wie NOT, AND, OR,...
  - Standard-Funktionen wie LOG, CHG,...

Beispiele: A B C D E F real Variable  
 A B C D E \$ string Variable  
 A B C D E % integer Variable  
 A B C 1  
 A 1 B C  
 A \$

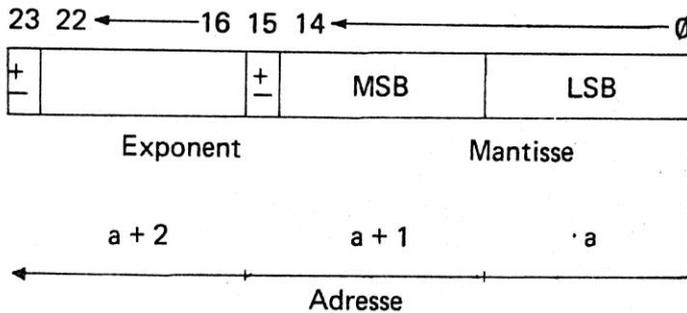
## Zahlen, Zeichen und Variablen

Numerische Werte werden intern als Gleitkomma-Werte (Real) dargestellt. Jeder Wert belegt im Speicher 3 Byte, davon nehmen

- die Mantisse 2 Byte u.
- der Exponent 1 Byte (zur Basis 2) ein.

## Gleitkomma-Zahlen

Darstellung von Gleitkommazahlen im Speicher:



15 Bits sind für die Mantisse reserviert, 1 Bit für das Mantissenvorzeichen; 7 Bits sind für den Exponenten reserviert, 1 Bit für das Exponentenvorzeichen. Für die Gleitkommazahlen ergibt sich eine Genauigkeit von 4.5 bis 4.8 Dezimalstellen.



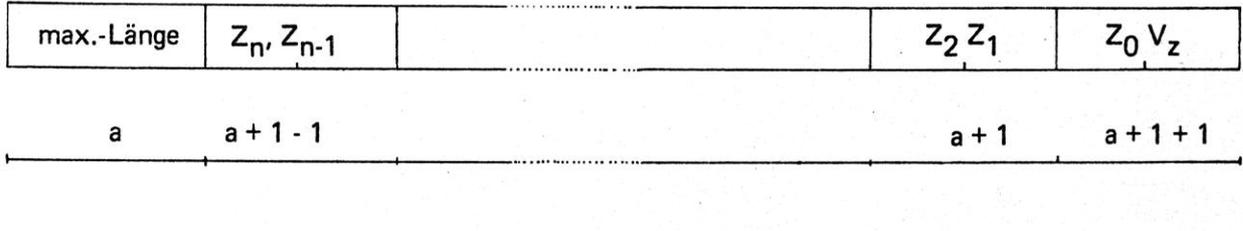
Festkomma-Zahlen werden intern als Strings von gepackten BCD-Zahlen dargestellt und belegen einschließl. des Vorzeichens max. 15 Bytes, d. h. max. können 29 Dezimalziffern gespeichert u. verarbeitet werden.  
 Folgende Vorzeichenverschlüsselungen sind zugelassen:

**Festkomma-Zahlen**

Binär	sedezimal	Vorzeichen	Binär	sedezimal	Vorzeichen
1010	A	+	1011	B	-
1100	C	+	1101	D	-
1110	E	+			
1111	F	+			

Darstellung von BCD-Strings im Speicher:

a = Adresse

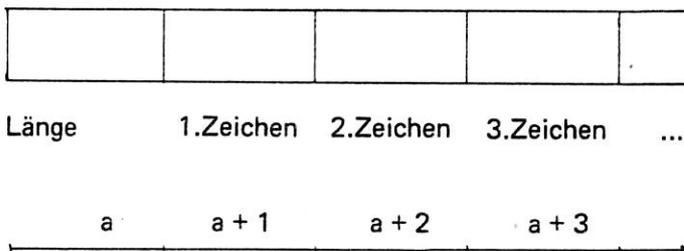


**String's**

Zeichenketten werden als String bezeichnet und im 7 Bit-ASCII-Code dargestellt, wobei jedes Zeichen 1 Byte belegt. Die max. Länge eines Strings können 128 Zeichen sein.

Darstellung von Strings im Speicher:

a = Adresse





Zahlenvariablen bezeichnen Speicherplätze, die numerische Worte beinhalten. Es wird unterschieden zwischen

- einfachen Zahlenvariablen
- indizierten Zahlenvariablen

### Zahlenvariable

Einfache Zahlenvariablen werden durch einen Namen identifiziert und repräsentieren stets nur einen Wert.

### Einfache Zahlenvariable

Beispiel:

A            Der hierfür benötigte Speicherplatz wird  
OTTO % durch den Basic-Compiler bei Umwand-  
X %        lung in den Objektcode reserviert.

Indizierte Zahlenvariablen sind Bestandteil eines ein- oder zweidimensionalen Feldes, d. h. eine Variable repräsentiert eine ganze Reihe von Werten.

### Indizierte Zahlenvariable

Beispiel:

A (0)            einfach indizierte Zahlenvariablen  
B (6 \* 7)  
Z (X)

NZ (E + 4, L) zweifach indizierte Zahlenvariablen  
Y (1, 2)  
ARRAY (A (3), X + 8)

- Der angegebene Variablen-Name bezeichnet ein Feld, das in einer DIM- bzw. CHAR-Anweisung reserviert wurde.
- Die Indizes beginnen stets mit 0, d. h. die jeweils erste Position hat den Index 0.
- Bei zweifacher Indizierung gibt der erste Index die Zeile und der zweite Index die Spalte an.
- Für Festkomma-Variablen ist nur eine einfache Indizierung zulässig. Die erforderliche Stellenzahl ist in einem sog. INTEGER-Statement anzugeben; fehlt diese Angabe, wird mit 7 Dezimalstellen gerechnet.



Stringvariablen werden durch Namen gekennzeichnet, welche an der letzten Stelle ein \$ - Zeichen haben, z. B.:

A\$  
TEXT\$  
M3\$

### Stringvariablen

- Der für eine Stringvariable erforderliche Speicherplatz kann mittels einer CHAR-Anweisung und der dazugehörigen Längenangabe reserviert werden.
- Wurde für eine Stringvariable keine CHAR-Anweisung gegeben, so wird automatisch ein Platz für 7 Zeichen reserviert.
- Für Zahlen- und Stringvariablen ist es möglich, ein und denselben Namen zu wählen, da Stringvariablen stets durch das \$-Zeichen gekennzeichnet sind. A und A\$ sind somit unterschiedliche Variablen.



alphaTronic

# BASIC COMPILER

Programm Befehle

## Deklarationsanweisungen

Reserviert für ein oder mehrere ein- oder zweidimensionale Zahlenfelder den entsprechenden Speicherplatz. Für jede Zahl werden 3 Byte benötigt.

**DIMZ (p)**  
**DIM z(p,q)**

Beispiel:

```
DIM A (9)
DIM B (3,2), x (100)
```

Belegt für ein oder mehrere Stringvariablen den erforderlichen Speicherplatz, wobei i die Länge des Strings und n die Anzahl der Elemente definiert, d. h. es ist eine einfache Indizierung möglich.

**CHAR s(i,n)**

Beispiel:

```
CHAR B$(20), C$(10)
CHAR D$(10, 20), E$(5,10)
```

Dieses Statement belegt den für Festkomma-Variablen (BCD-Stringvariable) erforderlichen Speicherplatz und gibt die Anzahl der Dezimalstellen und i die Anzahl der Feldelemente an, d. h. für B C D-Stringvariablen ist eine einfache Indizierung möglich.

**INTEGER b(s,i)**

Beispiel:

```
INTEGER B%(4), C%(10)
INTEGER D%(4, 10)
```

Im Statement LET wird einer Variablen ein Wert zugewiesen

**LET**

```
z. B. LET A = 5.1, B = 6.80 + C
      LET X 1$ = "TEXT"
      LET OTTO (1,2) = X/Y * N
      LET A% = 1 - (2 + (3 - (4 + (5 - (6 + 7))))))
```

Die rechte Seite des Gleichheitszeichens wird zuerst berechnet und das Ergebnis der linken Seite zugewiesen.



# BASIC COMPILER

## Programm-Befehle

### Organisatorische Anweisungen

Absolute Sprunganweisung, z. B. 10 GOTO 100

**GOTO n**

Bewirkt den Sprung in ein Unterprogramm, welches mit der Anweisungs-Nr. n beginnt. Nach dem Rücksprung aus dem Unterprogramm (durch RETURN) wird mit der nächstfolgenden Anweisungs-Nr. das Programm fortgesetzt, z. B. 10 GOSUB 100

**GOSUB n**

Rücksprung aus einem Unterprogramm.

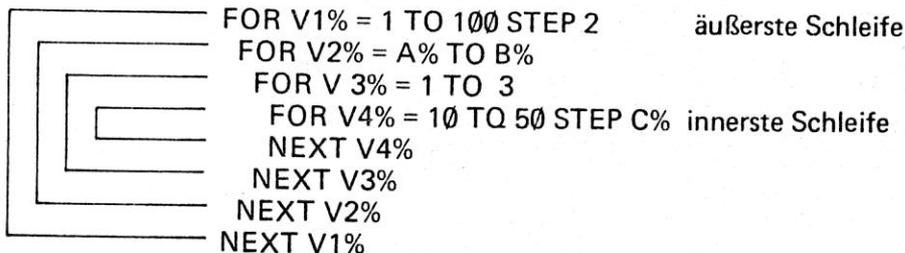
**RETURN**

Zum Definieren und Steuern einer Programmschleife.

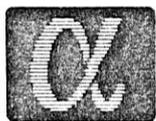
Z. B. FOR v = a TO e STEP s  
FOR v = a TO e  
NEXT v

**FOR....  
NEXT**

Die Schleife wird mit FOR eröffnet und mit NEXT abgeschlossen. Der Wert a bestimmt den Startwert des Laufindex, e den Endwert und s die Schrittweite. Die Angabe von s kann wahlweise erfolgen. Wird s nicht angegeben, so wird 1 angenommen. Negative Schrittweiten sind nicht zulässig.



- Zulässige Schachtelungen s. o., überkreuzte Schachtelungen führen zu Fehlern.
- Programmschleifen dürfen max. 8-fach verschachtelt sein.



alphatronic

# BASIC COMPILER

## Programm-Befehle

### Organisatorische Anweisungen IF THEN n

Dieses Statement erlaubt eine sog. bedingte Verzweigung, d. h. trifft der Ausdruck e zu, so wird ein Sprung zum Statement n durchgeführt, ansonsten wird die dem IF-Statement folgende Anweisung ausgeführt.

z. B. IF A = B THEN 70  
IF B/D AND A = C THEN 200

---

Zur besseren Übersichtlichkeit von Programmen ist es oft erforderlich, Bemerkungszeilen einzufügen,  
z. B. REM \*\*DIES IST EINE BEMERKUNG\*\*

**REM**

---

Kennzeichnet das logische Ende eines Programmes. Nach Ausführung der END-Anweisung kehrt das System in den KOMMANDO-Modus zurück und meldet sich mit der Nachricht READY.

**END**



alphaTronic

# BASIC COMPILER

## Programm-Befehle

### Arithmetische Operatoren

		Beispiele:
+	Addition	A + B + 3.4
-	Subtraktion	6.5 - C
*	Multiplikation	X * Y
/	Division	C / 2.0
**	Potenzierung	(2.0 ** 3.0)
		entspricht 2 <sup>3</sup>
MIN	Minimum	A MIN B
		Wert:
		A wenn A < B
		B wenn A ≥ B
MAX	Maximum	A MAX B
		Wert:
		A wenn A ≥ B
		B wenn A < B

### Vergleichs-Operatoren

=	gleich	IF A = B THEN 50
>	größer	IF A > B THEN 50
>=	größer, gleich	IF A >= B THEN 50
<	kleiner	IF A < B THEN 50
<=	kleiner, gleich	IF A <= B THEN 50
<>	ungleich	IF A <> B THEN 50

### Logische Operatoren

Das Ergebnis von logischen Verknüpfungen kann stets nur 0 oder 1 sein.

AND	Konjunktion	C AND B
		Wert:
		0 wenn A = 0 oder B = 0
		1 wenn A ≠ 0 und B ≠ 0
OR	Disjunktion	A OR B
		Wert:
		0 wenn A = 0 und B = 0
		1 wenn A ≠ 0 oder B ≠ 0
XOR	Exclusive Disjunktion	A XOR B
		Wert: 0 wenn A = B
		1 wenn A ≠ B
NOT	Negation	NOT A
		Wert: 0 wenn A ≠ 0
		1 wenn A = 0





Das Statement PRINT dient zur Ausgabe von Zahlenwerten und Texten auf einem Ausgabegerät. Die auszugebenden Daten stehen in einer sog. Argumentenliste hinter PRINT.

Folgende Argumente können vorkommen:

Zahlenkonstanten	Logische Ausdrücke
Zahlenvariablen	Vergleichsausdrücke
Zahlenvariablen indiziert	String-Konstanten (Texte)
Arithmetische Ausdrücke	String-Variablen

Alle Argumente sind durch ein Komma voneinander zu trennen. Normalerweise wird nach der Ausgabe aller Argumente ein Wagenrücklauf und Zeilenvorschub durchgeführt. Dies kann unterdrückt werden, indem man hinter das Argument ein Komma setzt.

Beispiel:

Basic-Compiler laden

@ NEW

10 LET I% = 5

20 PRINT 25, "IST GLEICH", I%\*\*2

30 END

@ START

25 IST GLEICH 25

@ NEW

10 CHAR A\$ (8)

20 LET A\$ = "COMPUTER"

30 PRINT A\$, A\$, "b b b", A\$

40 END

@ START

**Ein-Ausgabe**  
**PRINT a, a1, a2**



alphaTronic

# BASIC COMPILER

Programm Befehle

## Ein-Ausgabe PRINT DEV(d)

Normalerweise erfolgt die PRINT-Ausgabe auf dem Bildschirm. Soll ein anderes Ausgabegerät benutzt werden, so ist dessen Device-Nr. durch die u. a. Formel und nachfolgende Tabelle zu ermitteln.

Siehe OPEN-Befehl!

$$d = 10 * g + s$$

1	2	3	4	g / s
Tastatur Display	Sys-In Sys-Out	Sys-In Display	frei 1 frei 2	∅ = Command I/O
Sys-In z. B. Diskette	frei 3	Tastatur	frei 4	1 = Reader
Sys-Out z. B. Diskette	frei 7	Display	frei 5	2 = Writer
Drucker	Display	Sys-Out	frei 6	3 = Lister

Das Steuerwort DEV (d) kann auch in INPUT-Anweisungen verwendet werden, wodurch die in einer PRINT-Anweisung vorgenommene Device-Zuordnung verändert wird.

Die Device-Zuordnung bleibt also für alle folgenden PRINT- und INPUT-Anweisungen solange gültig, bis ein neues Gerät spezifiziert wird.

Beispiel:

```

10 CALL OPEN (1.0, 0.0, 0.0)
20 INPUT DEV (1), A$, B$
30 PRINT DEV (0), A$, B$
40 PRINT DEV (3), A$, B$
50 CALL CLOSE (1.0, 0.0)
60 CALL OPEN (2.0, 1.0, 0.0)
70 PRINT DEV (2) A$, B$
80 CALL CLOSE (2.0, 2.0)

```

```

10 Eröffnen Sys-In, File 0, Laufwerk 0
20 Einlesen vom Reader (Sys-In)
30 Ausgeben auf den Bildschirm
40 Ausgeben auf den Drucker
50 Abschließen Sys-In
60 Eröffnen Sys-Out, File 1, Laufwerk 0
70 Ausgeben auf den Writer (Sys-Out)
80 Abschließen Sys-Out und schreiben von
  2 EOF-, bzw. Filemarken.

```



alphaTronic

# BASIC COMPILER

## Programm Befehle

### Ein-Ausgabe

Das Steuerwort TAB in einer PRINT-Anweisung bewirkt, daß von der augenblicklichen Position des Cursors oder Schreibkopfes aus die in e angegebene Anzahl von Leerstellen ausgegeben wird.

Z. B. PRINT P%, TAB (6), E%  
PRINT TAB (1\*B-6.0), S%, "TEXT"

### PRINT TAB(e)

Diese Anweisung dient dazu, den angegebenen Variablen über das eingestellte Eingabegerät Werte zuzuweisen.

Wird eine Eingabe von Tastatur gefordert, so meldet sich das Programm mit einem Doppelpunkt (:) auf dem Bildschirm und wartet auf die Eingabe. Sind in der INPUT-Anweisung mehrere Variablen angegeben, so sind die Eingabewerte durch ein Nicht-Ziffernzeichen voneinander zu trennen, z. B. durch ein Komma. Bei der Eingabe von Werten ist es oft sinnvoll, einen erklärenden Text vorher auszugeben. Dies kann auch in der INPUT-Anweisung erfolgen, indem der entsprechende Text in Anführungszeichen gesetzt wird ("TEXT...")

### INPUT v1,v2...

Beispiel:

```
@ NEW
10 CHAR A$ (10)
20 INPUT A$, B%
30 PRINT A$, "+", B%
40 GOTO 30
```

Diese Funktion ermittelt den Tastatur-Status, d.h. die Variable v ist nach Ausführung des CIS-Statements auf Null gesetzt, wenn keine Taste gedrückt wurde. Das Ergebnis ist ungleich Null, falls irgendeine Taste gedrückt wurde.

Das Programm stoppt, im Gegensatz zur INPUT-Anweisung, nicht beim CIS-Statement, sondern führt die nächste Anweisung aus.

Welche Taste gedrückt wurde, kann mit einer nachfolgenden INPUT-Anweisung ermittelt werden.

Beispiel:

```
50 IF CIS (00) <> 00 THEN 1000
```

```
-  
-  
-
```

```
1000 INPUT A$
```

### CIS(v)



## BASIC COMPILER

### Programm-Befehle

#### CALL ROUTINEN

Ein CALL-Aufruf dient zur Auslösung einer bestimmten Funktion des Computer über die dazu vorgesehene Befehlsfolge, z. B. von

- Hardware-Funktionen, etwa zur Datenein- oder Ausgabe
- Software-Funktionen, wie der Mitteilung einer erforderlichen Bedienungsmaßnahme durch den Bediener
- besonderen Routinen, wie eines Unterprogrammes durch den Unterprogrammaufruf CALL .... oder GOSUB .... RETURN

Mit diesem Aufruf wird das Systemgerät eröffnet, und es besteht die Möglichkeit, mit nachfolgender DEV-Funktion in der PRINT- oder INPUT-Anweisung von einer sequentiellen Datei zu lesen oder in eine solche Daten auszugeben.

Die Ausgabe der Daten erfolgt stets im ASCII-7Bit Code, wobei die Daten erst in den Blockpuffer, und – wenn dieser gefüllt ist – auf das Systemgerät ausgegeben werden. Die Größe des Blockpuffers beträgt 128 Zeichen.

A, B und C muß stets im Gleitkommaformat angegeben werden.

A = Geräte-Nr., 1.0 = SYSIN, 2.0 = SYSOUT

B = Datei-Nr., max. 255.0

C = Laufwerks-Nr., z. B. 0.0 = Drive 0  
10.0 = Drive 1  
20.0 = Drive 2

Beispiel:

1. Lesen von Diskette, File 0, Laufwerk 0  
10 CALL OPEN (1.0, 0.0, 0.0)  
20 INPUT DEV (1), A\$, B\$
2. Schreiben auf Diskette, File 1, Laufwerk 0  
10 LET A = 2.0  
20 LET B = 1.0  
30 LET C = 0.0  
40 CALL OPEN (A, B, C)

#### CALL OPEN (A,B,C)



# BASIC COMPILER

## Programm-Befehle

### CALL ROUTINEN

#### CALL CLOSE (A,D)

Der Abschluß eines eröffneten Systemgerätes erfolgt mittels CLOSE, wobei ein nur teilgefüllter Ausgabepuffer vorher noch ausgegeben wird.  
Wird SYSOUT abgeschlossen, so werden entsprechend D Filemarken geschrieben.

A = Geräte-Nr., siehe OPEN  
D = Anzahl Filemarken schreiben  
A und D ist stets im Gleitkommaformat anzugeben.

#### CALL FINP (A,B,K\$)

Bewirkt ab der augenblicklichen Cursorposition eine sogenannte formatierte Eingabe.

Übergabeparameter:

A = Eingabevariable (Gleitkomma-, String- oder BCD-Variable)  
B = max. Eingabelänge + 1 für Abschlußtaste (Gleitkommavari-  
riable)  
K\$ = "1", d. h. A% = BCD-Variable  
"2", d. h. A\$ = String-Variable  
"3", d. h. A = Gleitkomma-  
Variable

Rückgabeparameter:

A = eingegebene Daten ohne Abschlußzeichen  
B = tatsächliche Eingabelänge *ohne* Abschlußzeichen  
K\$ = letztes Zeichen, falls die vorgegebene Länge B erreicht wurde, ansonsten Abschlußtaste



# BASIC COMPILER

## Programm-Befehle

### CALL ROUTINEN

#### CALL FMT (Maske)

Ermöglicht eine formatierte Ausgabe von BCD-Zahlen über eine sogenannte MASKE. Sie kann als Textkonstante oder als Textvariable übergeben werden.

**MASKE** "F [ [Z] [S] ]" max. Länge von 255 Byte

F = Füllzeichen (beliebiger Code)

Z = Ø, entspricht Platzhalter für Ziffer oder Füllzeichen  
1, ... 8, entspricht Platzhalter für stets auszugebende Ziffer

9, bewirkt, daß erst ab der 1. gültigen Ziffer lt. Maske gedruckt wird.

S = Sonderzeichen (außer 0 - 9)

Sie werden aus dem Maskenfeld unverändert übernommen, falls bereits gültige Ziffern ausgegeben wurden (> 0), ansonsten durch das Füllzeichen ersetzt.

Als Vorzeichen können + oder - angegeben werden.

+ = beide Vorzeichen werden ausgegeben, je nach Wertigkeit

- = es wird nur - ausgegeben, + wird durch blank ersetzt. Wurde weder + noch - ausgegeben, so wird der Betrag ohne Vorzeichen ausgegeben.

#### Beispiel

CALL FMT ("\* ØØ.ØØ1;ØØ-")



alphaTronic

## BASIC COMPILER

Programm-Befehle

### CALL ROUTINEN

#### CALL SYSIO ( A\$, L, F )

Ermöglicht den direkten Durchgriff auf den Systemgerätetreiber, was allerdings dessen genaue Kenntnis voraussetzt (siehe MOS-Manual)

- A\$ = Puffer (String), führt auf Treiberparameter Reg. BC
- L = max. Länge oder Kontrollwert, führt auf Reg. DE, bzw. nur E
- F = Funktion, führt auf Register 1

#### Beispiel

CALL SYSIO (A\$, 0.0, 10.0) = Rewind Systemgerät

LET L = 256.0  
CALL SYSIO (A\$, L, 128.0) = Lesen von max. 256 Byte nach A\$. Nach dem CALL-Aufruf steht in L die gelesene Blocklänge.

---

Byteweise Ein-/Ausgabe über Ports und Speicher. Ist die Portadresse größer als 255, so wird diese als absolute Speicheradresse betrachtet.

z. B. LET A\$ = INCW (PORT)  
LET OUTCW (PORT) = A\$

**INCW/PUTCW**



alphatronik

## BASIC COMPILER

Programm-Befehle

### CALL ROUTINEN

#### CALL MOVE (A, B%, C%)

Erlaubt den Transfer von Speicherinhalten

A = Variablenadresse

B% = Länge und Kennung

B% > 0, d. h. speichere A → C in Länge B%

B% < 0, d. h. speichere C → A in Länge B%

B% = 0, unzulässig

C% = absolute Speicheradresse

Beispiel: CALL MOVE (A, 3, 32768)

Die Variable A wird mit den Inhalten von Adr. 8000H, 8001H und 8002H geladen.

INTEGER A% (17)

CALL MOVE (A% - 10, 25344)

Der durch A% adressierte Speicherbereich wird (incl. Längenbyte von A%) nach 6300H bis 6309H geladen.

---

#### CALL FIC (A\$, WERT)

Belegung eines ganzen Strings mit einem Wert. Nach dem Start eines Programmes sind alle Stringelemente mit binär Null belegt. Durch den Aufruf CALL FIC (A\$, WERT) kann ein String in allen Elementen mit dem angegebenen Wert belegt werden, wobei WERT als Gleitkommazahl des gewünschten Zeichens anzugeben ist, z. B. 32.0 für blank (siehe Anhang A, ASCII-Tabelle).



alphaTronic

## BASIC COMPILER

Programm-Befehle

Transfer von Stringelementen.

- $C = 0.0$  d.h. Übertragen der definierten Stringlänge von A\$ in das 1. Byte von B\$.
- $C > 0.0$  d.h. Übertragen eines Stringelementes von A\$ nach B\$. Das C-te Element von A\$ wird in das 1. Element von B\$ übertragen.
- $C < 0.0$  d.h. Übertragen des 1. Elementes von B\$ in das C-te Element von A\$.
- C ist stets im Gleitkommaformat anzugeben.

### CALL ROUTINEN

#### CALL DAS(A\$, B\$, C)

Umwandlung von Strings in BCD-Zahlen und umgekehrt

- $C > 0.0$  d.h. String in BCD in der Länge C von rechts nach links umwandeln.
- $C < 0.0$  d.h. BCD in String in der Länge C von rechts nach links umwandeln.
- $C = 0.0$  d.h. String in BCD in der Länge von B\$ von rechts nach links wandeln.
- C ist stets im Gleitkommaformat anzugeben.

#### CALL CSB(A%, B\$, C)

Runden einer BCD (INTEGER)-Zahl

- A% = BCD-Zahl  
B = Rundungsstelle (Gleitkommaformat)

Auf die (B-1)-te Stelle von A% (von rechts gezählt) wird eine 5 addiert und bis zur letzten Stelle NULLEN eingespeichert.

#### CALL BIFX(A%, B)

Beispiel:

- ```
10 INPUT A%, B
20 CALL BIFX(A%, B)
30 PRINT A%
40 END
```



## BASIC COMPILER

Programm Befehle

Mit Hilfe des CALL-Statements ist es möglich, bereits übersetzte ASSEMBLER, BASIC oder PL/M-Subroutines innerhalb eines BASIC-Programmes aufzurufen, wobei bis zu 3 Parameter übergeben werden können.

### CALL ROUTINEN

**CALL name(x1, x2, ...)**

Um ein Basic-Programmstück als Unterprogramm zu deklarieren, ist dieses Statement zu Beginn des Programmes anzugeben.

### SUBROUTINE

Beispiel:

```
10 SUBROUTINE
20 INTEGER 1% (9), GELD% (6,49)
30 CHAR NAM$ (6,49)
```

```
—
—
—
—
—
```

```
180 RETURN
```

Der Rücksprung aus dem Unterprogramm in das Hauptprogramm erfolgt mit RETURN-Statement. Die Übergabe von beliebig vielen Parametern kann durch eine identische Deklaration der Variablen im Unterprogramm und im Hauptprogramm erfolgen.

### RETURN

Um in einem BASIC-Programm eine externe Referenz, d. h. einen Verweis auf ein Unterprogramm, zu erzeugen, wird das Statement EXTERNAL benutzt.

### EXTERNAL Name (Adr.)

Name = Symbolischer Name des Unterprogrammes  
(Adr.) = Angabe der Adresse (dezimal), auf welche sich der symbolische Name bezieht.

Beispiel:

```
10 EXTERNAL SPRINT (24832)
20 INTEGER 1% (9), GELD% (6,49)
30 CHAR NAM$ (6,49)
40 INTEGER SUM% (13), MWST% (S)
```

```
—
—
—
—
—
```

```
120 CALL SPRINT
```

```
—
—
—
—
—
```

```
980 END
```



# BASIC COMPILER

## Fehlermeldungen und Fehlercodes

### Warnungen

Auf angegebene Variable erfolgt kein Zugriff 1  
Keine Zuweisung an angegebene Variable 2

Beispiel

```
CS  WARNING 2
ISA  xxxxH xxxxH
VSA  xxxxH xxxxH
```

Diese Fehler-Nummern können im BASIC-Programm bei Verwendung der Variablen ERR direkt abgefragt werden.

### Fehlermeldungen der Library

Beispiel

```
10 IF ERR = 5.0 THEN 20
20 PRINT ,ERR=", ERR
```

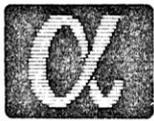
### BEDEUTUNG

### Fehler-Nr.

|                                                       |    |
|-------------------------------------------------------|----|
| Überlauf                                              | 1  |
| Längenfehler (nur BCD Strings)                        | 2  |
| Divisor ist Null                                      | 3  |
| Operand ist unnormalisierte Gleitkommazahl            | 6  |
| Falsche Eingabe im Rahmen des INPUT-Statements        | 8  |
| Negatives Argument bei logarithmischen Funktionen     | 10 |
| Argument bei logarithmischen Funktionen ist Null      | 11 |
| Argument bei trigonometrischen Funktionen ist zu groß | 12 |
| Überlauf Index                                        | 13 |
| Datenfehler                                           | 30 |
| Gerät nicht bereit (allgem. Fehler)                   | 31 |
| Datenträgerende                                       | 32 |
| Dateiende (Filemark)                                  | 33 |

Beispiel:

```
I/O 33
```



alphaTronic

# BASIC COMPILER

## Fehlermeldungen und Fehlercodes

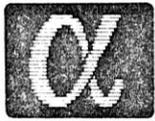
### Fehlercodes der Analysephase

| BEDEUTUNG                                     | CODE |
|-----------------------------------------------|------|
| Zeilennummer fehlt                            | 01   |
| Zeilennummer zu groß                          | 02   |
| Zeile endet nach der Zeilennummer             | 03   |
| BASIC-Anweisung fehlt oder ist unzulässig     | 04   |
| Zeile ist zu lang (mehr als 69 Zeichen)       | 05   |
| Zeile enthält zu viele Fehler                 | 06   |
| Zeilenende vorzeitig gefunden (unvollständig) | 07   |
| Information nach Anweisungsende               | 08   |
|                                               |      |
| Name mehrfach definiert                       | 10   |
| Variablenname fehlt                           | 11   |
| Variablenname ist unzulässig                  | 12   |
| Feldvereinbarung ist unzulässig               | 13   |
| Zeichen ist unzulässig; Komma fehlt           | 14   |
| Teilstrings sind nicht erlaubt                | 15   |
| Marke fehlt oder ist unzulässig               | 16   |
| Laufparameter muß einfache Variable sein      | 17   |
|                                               |      |
| Element noch nicht implementiert              | 20   |
| Fehler in Klammerstruktur                     | 21   |
| Operator hier unzulässig                      | 22   |
| Aufbau des Ausdrucks unzulässig               | 23   |
| Index- oder Parameteranzahl ist falsch        | 23   |
| Zahlenkonstante falsch oder unzulässig        | 25   |
| Keine Zuweisung im LET                        | 26   |
| Reserviertes Wort hier unzulässig             | 27   |
| Komma hier unzulässig                         | 28   |
|                                               |      |
| Abbruch wegen Speichermangel                  | A1   |
| Abbruch wegen Überlauf der Symboltabelle      | A2   |
| Abbruch wegen Überlauf der Sprungzieltabelle  | A3   |
| Abbruch wegen fehlerhafter Quelleingabe       | A4   |
| ISA innerhalb VSA                             | A5   |

Beispiel

40 PRINT A\$; B

14 ← Fehlerhinweis  
      ? , ← Fehler-Nr.



alphaTronic

# BASIC COMPILER

## Fehlermeldungen und Fehlercodes

### Fehlercodes der Generierungsphase

| BEDEUTUNG                                                                                                     | CODE |
|---------------------------------------------------------------------------------------------------------------|------|
| Leistung noch nicht implementiert                                                                             | C0   |
| Marke in der Sprungzieltabelle mehrfach vorhanden                                                             | C1   |
| Unbekanntes Sprungziel                                                                                        | C2   |
| Zu viele Feldelemente in einer DIM-Anweisung                                                                  | C3   |
| Mixed-mode Fehler<br>oder<br>Länge einer Stringvariablen für numerische Zuweisung<br>zu klein                 | C4   |
| Speichermangel während der Generierungsphase                                                                  | C5   |
| Unzulässige Schachtelung von FOR-Schleifen<br>oder<br>FOR-Anweisung ohne NEXT                                 | C6   |
| ONINT-Statement mehrfach mit gleicher Interrupt-<br>Nummer vorhanden oder Interrupt-Nummer keine<br>Konstante | C7   |
| Unerlaubte Stellung von Prozess- I/O-Variablen                                                                | C8   |
| Operator hier unzulässig                                                                                      | C9   |



alphaTronic

**BUS**  
Pin Belegung

Die dritte Reihe der 96 pol. Peripheriestecker ist über alle 32 Kontakte frei für die jeweiligen Sonder-Ein/Ausgänge der einzelnen Leiterplatte.

### MC-80-Verdrahtungsfelder

Diese dritte Reihe beinhaltet keine MC 80-System-Spannungsanschlüsse.

Unter Systemspannung wird verstanden:

+ 5 V, ± 12 V

Auf der dritten Reihe können z.B. Motorspannungen (Floppy, Cassette) liegen.

Reihe a

Reihe b

Reihe c (Option für Peripherie-Abgänge)

### Belegung MC 80 - BUS

|    |               |    |               |    |   |
|----|---------------|----|---------------|----|---|
| 1  | + 5 V         | 33 | + 5 V         | 65 | x |
| 2  | OV            | 34 | OV            | 66 | x |
| 3  | + 12 V        | 35 | + 12 V        | 67 | x |
| 4  | OV            | 36 | OV            | 68 | x |
| 5  | - 12 V        | 37 | - 12 V        | 69 | x |
| 6  | OV            | 38 | OV            | 70 | x |
| 7  |               | 39 | UHR           | 71 | x |
| 8  | <u>MPR</u>    | 40 |               | 72 | x |
| 9  | <u>ADR 0</u>  | 41 | <u>MEMR</u>   | 73 | x |
| 10 | <u>ADR 1</u>  | 42 | <u>MEMW</u>   | 74 | x |
| 11 | <u>ADR 2</u>  | 43 | <u>IOR</u>    | 75 | x |
| 12 | <u>ADR 3</u>  | 44 | <u>IOW</u>    | 76 | x |
| 13 | <u>ADR 4</u>  | 45 | <u>INTA</u>   | 77 | x |
| 14 | <u>ADR 5</u>  | 46 | <u>RESET</u>  | 78 | x |
| 15 | <u>ADR 6</u>  | 47 | <u>RST</u>    | 79 | x |
| 16 | <u>ADR 7</u>  | 48 |               | 80 | x |
| 17 | <u>ADR 8</u>  | 49 | <u>WAIT</u>   | 81 | x |
| 18 | <u>ADR 9</u>  | 50 | <u>HOLD</u>   | 82 | x |
| 19 | <u>ADR 10</u> | 51 | <u>HOLDA</u>  | 83 | x |
| 20 | <u>ADR 11</u> | 52 | <u>BUSEN</u>  | 84 | x |
| 21 | <u>ADR 12</u> | 53 | <u>0 2</u>    | 85 | x |
| 22 | <u>ADR 13</u> | 54 | <u>STST B</u> | 86 | x |
| 23 | <u>ADR 14</u> |    |               | 87 | x |
| 24 | <u>ADR 15</u> | 56 | <u>INT</u>    | 88 | x |
| 25 | <u>DAT 0</u>  | 57 | <u>IR 0</u>   | 89 | x |
| 26 | <u>DAT 1</u>  | 58 | <u>IR 1</u>   | 90 | x |
| 27 | <u>DAT 2</u>  | 59 | <u>IR 2</u>   | 91 | x |
| 28 | <u>DAT 3</u>  | 60 | <u>IR 3</u>   | 92 | x |
| 29 | <u>DAT 4</u>  | 61 | <u>IR 4</u>   | 93 | x |
| 30 | <u>DAT 5</u>  | 62 | <u>IR 5</u>   | 94 | x |
| 31 | <u>DAT 6</u>  | 63 | <u>IR 6</u>   | 95 | x |
| 32 | <u>DAT 7</u>  | 64 | <u>IR 7</u>   | 96 | x |



# BUS Signal-Beschreibung

## Kurzbeschreibung der BUS-Signale

|                                                                                                                                       |  |                                          |
|---------------------------------------------------------------------------------------------------------------------------------------|--|------------------------------------------|
| Adressleitung ADR $\emptyset$ - 15 (64 K Byte)                                                                                        |  | <b>1. ADR<math>\emptyset</math> - 15</b> |
| Datenleitungen DAT $\emptyset$ - 7                                                                                                    |  | <b>2. DAT <math>\emptyset</math> - 7</b> |
| Memory-Read; (spätestens 200 ns nach der abfallenden Flanke müssen die Daten stabil sein) gilt ebenso für MEMW, IOR, IOW, INTA        |  | <b>3. MEMR</b>                           |
| Memory-Write                                                                                                                          |  | <b>4. MEMW</b>                           |
| Input-Output-Read                                                                                                                     |  | <b>5. IOR</b>                            |
| Input-Output-Write <span style="margin-left: 20px;">benutzen ADR 0-7</span>                                                           |  | <b>6. IOW</b>                            |
| Interrupt-Acknowledged                                                                                                                |  | <b>7. INTA</b>                           |
| Reset; General Null (PC - 0000)                                                                                                       |  | <b>8. RESET</b>                          |
| Restart; mit $\emptyset$ 2-getaktetem Reset PHI                                                                                       |  | <b>9. RST</b>                            |
| Ready; CPU geht in T <sub>W</sub> -Status muß > 120 ns vor negativer $\emptyset$ 2 Flanke auf OV gehen.                               |  | <b>10. RDY</b>                           |
| Wait; Rückmeldung von der CPU, daß sie im T <sub>W</sub> -Status ist. (Nur bei CPU I)                                                 |  | <b>11. WAIT</b>                          |
| Hold; CPU geht in den Hold-Status (für DMA).                                                                                          |  | <b>12. HOLD</b>                          |
| Hold-Acknowledged; CPU meldet zurück, daß sie sich im Hold-Status befindet.                                                           |  | <b>13. HOLDA</b>                         |
| BUS-Enable; alle BUS-Ausgänge gehen in den Tri-State-Zustand bei BUSEN. (Nur bei CPU I)                                               |  | <b>14. BUSEN</b>                         |
| Takt $\emptyset$ 2 der CPU; 2.048 MHz                                                                                                 |  | <b>15. <math>\emptyset</math>2</b>       |
| Status-Strobe; abfallende Flanke taktet über CPU-8085 A-Daten-Bus die Status-Information in den BUS-Controller 8228. (Nur bei CPU I). |  | <b>16. STSTB</b>                         |
| Interrupt-Meldung zur CPU                                                                                                             |  | <b>17. INT</b>                           |
| Interrupt $\emptyset$ -7; abfallende Flanke triggert D-Flip-Flop in PICU 8259 und meldet Interrupt. (nur bei Interrupt Controller)    |  | <b>18. IRO-7</b>                         |



**CPU**  
Die CPU

## Allgemeines

Der Intelbaustein 8085 A bildet den Prozessor der CPU III-Baugruppe.

Für diese Baugruppe wurde ein Quarztakt von 4 000 MHz gewählt. Somit hat der herausgeführte Takt  $\Phi 2$  eine Frequenz von ca. 2 MHz.

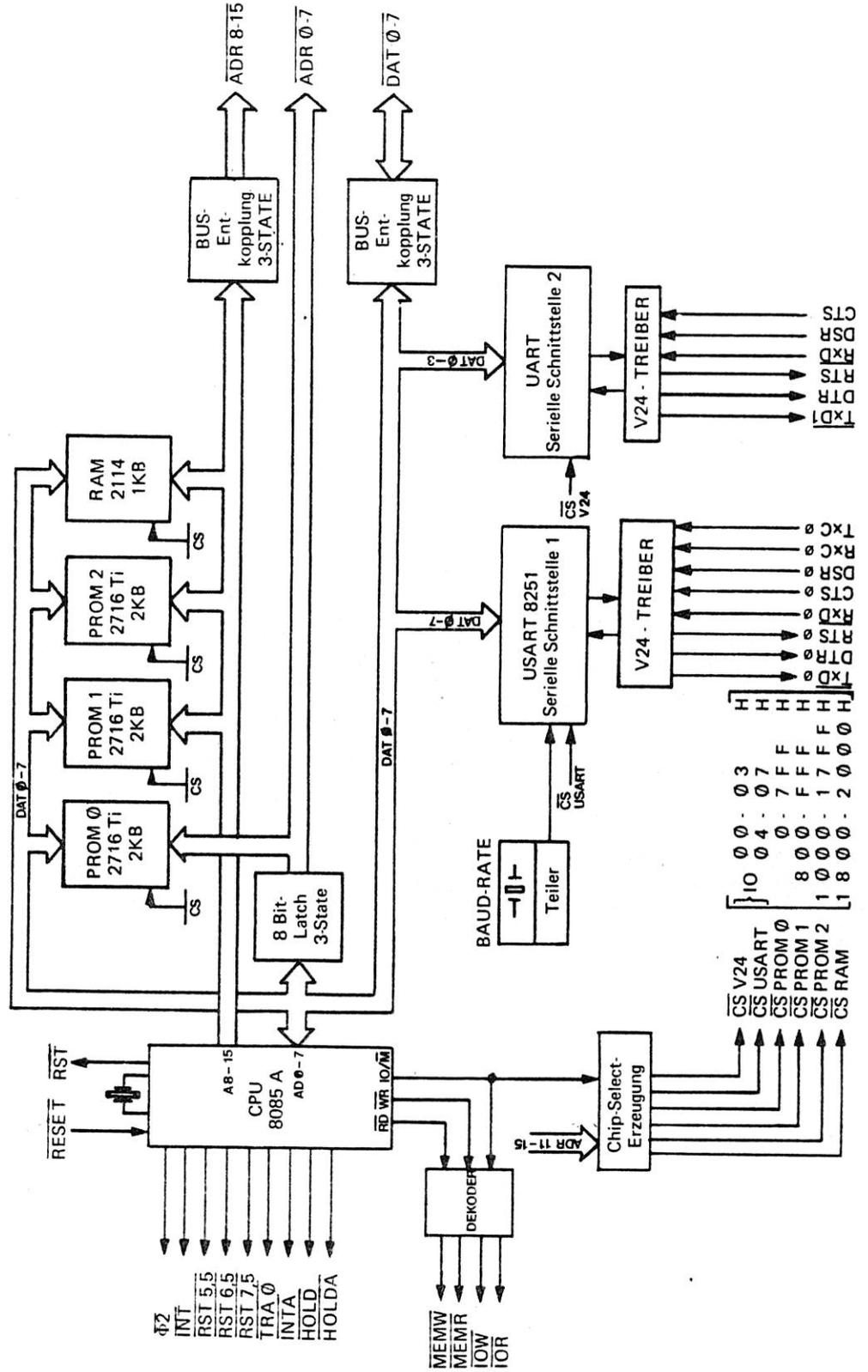
### 1. Prozessor

---

Die maximale Speicherausbaustufe ist wie folgt:

|        |       |            |
|--------|-------|------------|
| 6KByte | PROM; | 2716 Texas |
| 1KByte | RAM;  | 2114 Intel |

### 2. Speicher





## **Serielle Schnittstelle**

Die CPU III besitzt zwei serielle Schnittstellen.

- a) Prozessor-Ein/Ausgang „SID; SOD“
- b) separater USART

Durch Einsatz des Prozessors 8085 stehen dem Benutzer der Eingang „SID“ und der Ausgang „SOD“ zur Übertragung von seriellen Daten zur Verfügung. Zusätzlich können verschiedene Status-Leitungen gesetzt bzw. abgefragt werden. (DSR; CTS; DTR; RTS).

Nach- bzw. vorgeschaltet sind, entsprechend den RS 232-Empfehlungen, Treiber- bzw. Empfängerbausteine.

**SID;SOD**

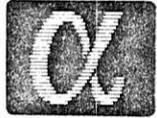
Die zweite serielle Schnittstelle ist mit dem USART 8251 Intel aufgebaut.

Folgende Ein/Ausgänge sind mit Treiber-Bausteinen versehen:

RxC; TxC; RxD; TxD; DSR; DTR; CTS; RTS

Die Schnittstelle entspricht ebenfalls den RS 232 C-Empfehlungen.

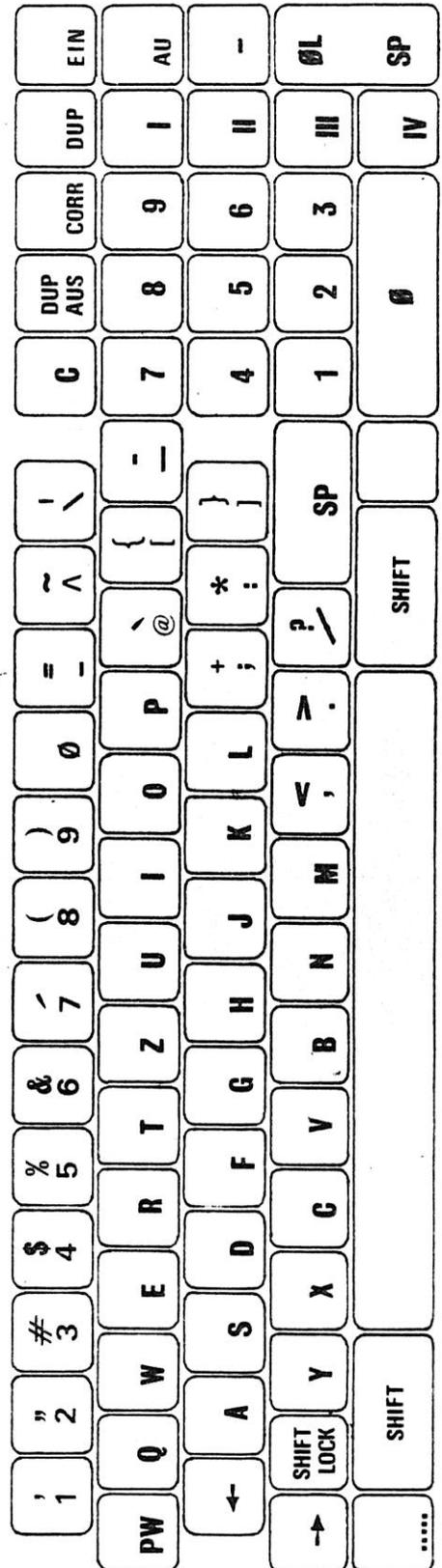
**USART**



alphatronic

# TASTATUR

Die Tastatur



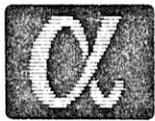


# TASTATUR

## Der Tastencode

### Control-Codes und ASCII-Codes

| Code  | Zeichen/Taste              | Code | Zeichen/Taste        | Code | Zeichen/Taste                |
|-------|----------------------------|------|----------------------|------|------------------------------|
| 0     | PW-Taste                   | 63   | ? Fragezeichen       | 111  | o                            |
| 1     | ← Cursor links und I-Taste | 64   | @ Klammeraffe        | 112  | p                            |
| 2     | II-Taste                   | 65   | A                    | 113  | q                            |
| 3     | III-Taste                  | 66   | B                    | 114  | r                            |
| 4     | C-Taste und IV-Taste       | 67   | C                    | 115  | s                            |
| 5     | DUP/AUS-Taste              | 68   | D                    | 116  | t                            |
| 6     | CORR-Taste                 | 69   | E                    | 117  | u                            |
| 7     | DUP-Taste                  | 70   | F                    | 118  | v                            |
| 8     | I-Taste shift              | 71   | G                    | 119  | w                            |
| 9     | AU-Taste                   | 72   | H                    | 120  | x                            |
| 10    | II-Taste shift             | 73   | I                    | 121  | y                            |
| 11    | —Taste                     | 74   | J                    | 122  | z                            |
| 12    | III-Taste shift            | 75   | K                    | 123  | { Klammer auf                |
| 13    | SP-Taste innen             | 76   | L                    | 124  | Strich                       |
| 14    | IV-Taste shift             | 77   | M                    | 125  | } Klammer zu                 |
| 15    | SP-Taste außen             | 78   | N                    | 126  | ~ sinus                      |
| 16-31 | nicht belegt               | 79   | O                    | 127  | Taste<br>(nicht beschriftet) |
| 32    | Space                      | 80   | P                    |      |                              |
| 33    | ! Ausrufezeichen           | 81   | Q                    |      |                              |
| 34    | " Anführungszeichen        | 82   | R                    |      |                              |
| 35    | # Nummernzeichen           | 83   | S                    |      |                              |
| 36    | \$ Dollar-Zeichen          | 84   | T                    |      |                              |
| 37    | % Prozent                  | 85   | U                    |      |                              |
| 38    | & kaufm. UND               | 86   | V                    |      |                              |
| 39    | ' Apostroph                | 87   | W                    |      |                              |
| 40    | ( Klammer auf              | 88   | X                    |      |                              |
| 41    | ) Klammer zu               | 89   | Y                    |      |                              |
| 42    | * Stern                    | 90   | Z                    |      |                              |
| 43    | + plus                     | 91   | [ Klammer auf        |      |                              |
| 44    | , Komma                    | 92   | \ Schrägstrich links |      |                              |
| 45    | - minus                    | 93   | ] Klammer zu         |      |                              |
| 46    | . Punkt                    | 94   | ^                    |      |                              |
| 47    | / Schrägstrich rechts      | 95   | _ Unterstreichung    |      |                              |
| 48    | 0                          | 96   | ` Apostroph          |      |                              |
| 49    | 1                          | 97   | a                    |      |                              |
| 50    | 2                          | 98   | b                    |      |                              |
| 51    | 3                          | 99   | c                    |      |                              |
| 52    | 4                          | 100  | d                    |      |                              |
| 53    | 5                          | 101  | e                    |      |                              |
| 54    | 6                          | 102  | f                    |      |                              |
| 55    | 7                          | 103  | g                    |      |                              |
| 56    | 8                          | 104  | h                    |      |                              |
| 57    | 9                          | 105  | i                    |      |                              |
| 58    | : Doppelpunkt              | 106  | j                    |      |                              |
| 59    | ; Semikolon                | 107  | k                    |      |                              |
| 60    | < kleiner                  | 108  | l                    |      |                              |
| 61    | = gleich                   | 109  | m                    |      |                              |
| 62    | > größer                   | 110  | n                    |      |                              |



alphatronic

## TASTATUR INTERFACE

### MC 80-39

Das Tastatur-Interface

#### Technische Beschreibung:

Je Tastendruck wird ein UET-Signal an der Vorderflanke und anschließend ein CNTL-Signal an der Rückflanke des Tastatur-Strobe-Impulses erzeugt. Mit dem Signal UET wird der Tastencode, gefiltert und entprellt durch Schmitt-Trigger, in den Eingabepuffer eingelesen. Der im Strobed-Input-Mode betriebene 8279 übernimmt diese Information mit dem Signal CNTL. Falls die DAUER-Taste gedrückt ist, erzeugen 2 Monoflops periodisch CNTL-Signale, so daß der Chip den jeweils letzten Tastencode in seinen FIFO-Speicher lädt. Nachdem der Code übernommen ist, erzeugt der 8279 ein Interruptsignal (INT $\emptyset$  bis 7 durch Lötbrücken wählbar, INT $\emptyset$  voreingestellt). Die CPU kann dann mit Hilfe dieses Signals oder durch Abfrage des FIFO-Status den Tastaturcode weiter verarbeiten.

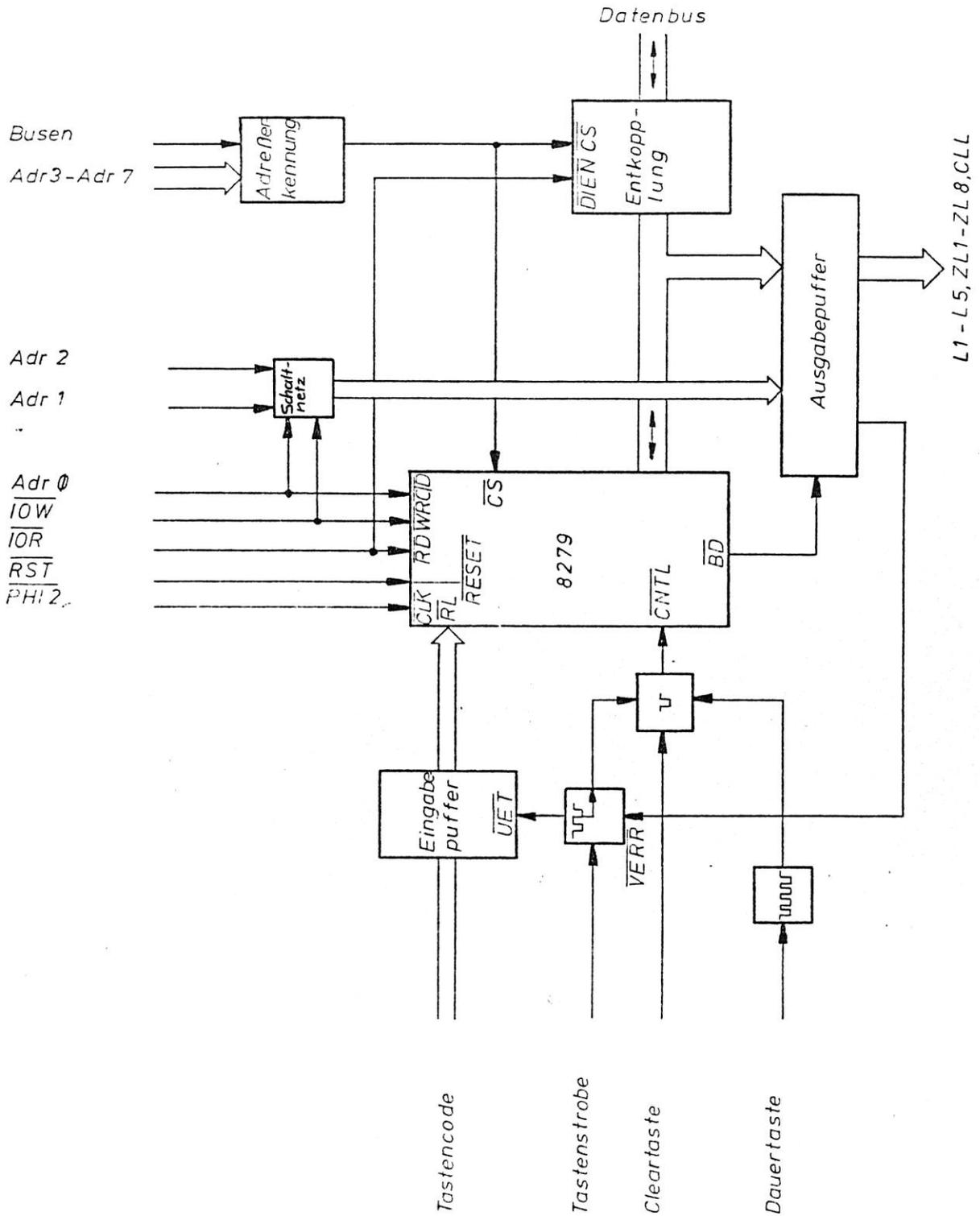
Die Baugruppe besitzt ferner 4 4-Bit-Ausgabepuffer. Diese können über 4 Output-Adressen geladen werden. Zusätzlich wird die Information im 8279 gespeichert, über den sie auch wieder rücklesbar ist. Ein Ausgabepuffer ist mit Treibern versehen, so daß Kontrollämpchen oder Beeps direkt angeschlossen werden können. 8 weitere Ausgabeleitungen mit TTL-Pegel stehen für Benutzeranwendungen zur Verfügung.

Stromversorgung: +5V  $\pm$ 5% typ.:0,4A max.:0,65A

| PIN | Bezeichnung | Diagramm                                                                            | Erläuterung   |
|-----|-------------|-------------------------------------------------------------------------------------|---------------|
| 01  | +5V         |                                                                                     |               |
| 02  | 0V          |                                                                                     |               |
| 03  |             |                                                                                     |               |
| 04  | 0V          |                                                                                     |               |
| 05  |             |                                                                                     |               |
| 06  | 0V          |                                                                                     |               |
| 07  |             |                                                                                     |               |
| 08  |             |                                                                                     |               |
| 09  | ADR 0       |    | Adresse-BUS 0 |
| 10  | ADR 1       |    | Adresse-BUS 1 |
| 11  | ADR 2       |    | Adresse-BUS 2 |
| 12  | ADR 3       |    | Adresse-BUS 3 |
| 13  | ADR 4       |    | Adresse-BUS 4 |
| 14  | ADR 5       |    | Adresse-BUS 5 |
| 15  | ADR 6       |    | Adresse-BUS 6 |
| 16  | ADR 7       |    | Adresse-BUS 7 |
| 17  |             |                                                                                     |               |
| 18  |             |                                                                                     |               |
| 19  |             |                                                                                     |               |
| 20  |             |                                                                                     |               |
| 21  |             |                                                                                     |               |
| 22  |             |                                                                                     |               |
| 23  |             |                                                                                     |               |
| 24  |             |                                                                                     |               |
| 25  | DAT 0       |  | Daten-BUS 0   |
| 26  | DAT 1       |  | Daten-BUS 1   |
| 27  | DAT 2       |  | Daten-BUS 2   |
| 28  | DAT 3       |  | Daten-BUS 3   |
| 29  | DAT 4       |  | Daten-BUS 4   |
| 30  | DAT 5       |  | Daten-BUS 5   |
| 31  | DAT 6       |  | Daten-BUS 6   |
| 32  | DAT 7       |  | Daten-BUS 7   |

| PIN | Bezeichnung | Diagramm                                                                            | Erläuterung        |
|-----|-------------|-------------------------------------------------------------------------------------|--------------------|
| 33  | +5VL        |                                                                                     | +5V Logiksp.       |
| 34  | 0VL         |                                                                                     | 0V Logikmasse      |
| 35  |             |                                                                                     |                    |
| 36  | 0VL         |                                                                                     | 0V Logikmasse      |
| 37  |             |                                                                                     |                    |
| 38  | 0VL         |                                                                                     | 0V Logikmasse      |
| 39  |             |                                                                                     |                    |
| 40  |             |                                                                                     |                    |
| 41  |             |                                                                                     |                    |
| 42  |             |                                                                                     |                    |
| 43  | IOR         |    | Input-Output Read  |
| 44  | IOW         |    | Input-Output Write |
| 45  |             |                                                                                     |                    |
| 46  | Reset       |    | Reset              |
| 47  |             |                                                                                     |                    |
| 48  |             |                                                                                     |                    |
| 49  |             |                                                                                     |                    |
| 50  |             |                                                                                     |                    |
| 51  |             |                                                                                     |                    |
| 52  | BUSEN       |    | BUS-Enable         |
| 53  | PHI 2       |    | CPU-Takt 0-2       |
| 54  |             |                                                                                     |                    |
| 55  |             |                                                                                     |                    |
| 56  | INT         |  | Interrupt          |
| 57  | IRO         |  | Interrupt 0        |
| 58  | IR1         |  | Interrupt 1        |
| 59  | IR2         |  | Interrupt 2        |
| 60  | IR3         |  | Interrupt 3        |
| 61  | IR4         |  | Interrupt 4        |
| 62  | IR5         |  | Interrupt 5        |
| 63  | IR6         |  | Interrupt 6        |
| 64  | IR7         |  | Interrupt 7        |

| PIN | Bezeichnung   | Diagramm | Erläuterung                 |
|-----|---------------|----------|-----------------------------|
| 65  | L3            |          | Lampe 3                     |
| 66  | L2            |          | Lampe 2                     |
| 67  | L1            |          | Lampe 1                     |
| 68  | CLL           |          | Clear Lampe                 |
| 69  | CLT           |          | Clear Taste                 |
| 70  | GN            |          | GN-Taste                    |
| 71  | FSTR          |          | Funktionstast.-Strobe       |
| 72  | α+10er Strobe |          | α+10er Strobe               |
| 73  | DT            |          | Dauertaste                  |
| 74  | INF 7         |          | Eingabe Inf. 2 <sup>7</sup> |
| 75  | INF 6         |          | Eingabe Inf. 2 <sup>6</sup> |
| 76  | INF 5         |          | Eingabe Inf. 2 <sup>5</sup> |
| 77  | INF 4         |          | Eingabe Inf. 2 <sup>4</sup> |
| 78  | INF 3         |          | Eingabe Inf. 2 <sup>3</sup> |
| 79  | INF 2         |          | Eingabe Inf. 2 <sup>2</sup> |
| 80  | INF 1         |          | Eingabe Inf. 2 <sup>1</sup> |
| 81  | INF 0         |          | Eingabe Inf. 2 <sup>0</sup> |
| 82  |               |          |                             |
| 83  |               |          |                             |
| 84  |               |          |                             |
| 85  | L4            |          | Lampe 4 (Beep)              |
| 86  |               |          |                             |
| 87  |               |          |                             |
| 88  |               |          |                             |
| 89  | ZL 1          |          | Zusatzlampe 1               |
| 90  | ZL 2          |          | Zusatzlampe 2               |
| 91  | ZL 3          |          | Zusatzlampe 3               |
| 92  | ZL 4          |          | Zusatzlampe 4               |
| 93  | ZL 5          |          | Zusatzlampe 5               |
| 94  | ZL 6          |          | Zusatzlampe 6               |
| 95  | ZL 7          |          | Zusatzlampe 7               |
| 96  | ZL 8          |          | Zusatzlampe 8               |





alphatronik

# TASTATUR INTERFACE MC 80-39

## Funktionsbeschreibung

Die Baugruppe kann über einen Wahlschalter 1 aus 16 Kanaladressen im Adreßraum  $\leq 127$  ( $2^7 = 0$ ) oder  $128 \leq$  Adreßraum  $\leq 255$  ( $2^7 = 1$ ) Adreßbit  $2^3 - 2^6$  belegen. ( $2^7$  durch Brücke umschaltbar)

### Adressen

Alle Tasten (-, 10er, Funktion, Clear) erzeugen je Tastendruck ein UET-Signal an der Strobe-Vorderflanke und ein CNTL-Signal an der Strobe-Rückflanke. Diese Signale werden durch 2 Monos erzeugt.

Mit dem Signal UET wird der Tastencode über einen Eingangsfilter in den Eingabepuffer übernommen.

Der im Strobed Input Mode betriebene Chip 8279 übernimmt diese Information mit dem Signal CNTL. Durch Drücken der Dauer-Taste, werden durch 2 Monos periodisch CNTL-Signale erzeugt, so daß der Chip den jeweils letzten Tastencode in sein FIFO lädt.

Durch Abfragen des FIFO-Status, oder durch Akzeptieren des vom Chip erzeugten Interrupt-Signales, kann der Tastencode vom Rechner über den Datenbus übernommen werden.

### Tastaturinformation

Die Baugruppe besitzt 4 x 4 Bit Ausgabepuffer (4 x 74 LS 175), die durch 4 verschiedene Adressen, in Verbindung mit IOW, geladen werden können. Diese Information wird zusätzlich im 8279 gespeichert, über den sie wieder rücklesbar ist.

Mit dem Signal VERR können alle Tastenstrobes, mit Ausnahme des Clear-Tasten-Strobes, verriegelt werden. Das Signal L 4 kann zur Ansteuerung einer weiteren Lampe oder eines Hup-Kurztones verwendet werden.

Alle Puffer können über das Signal  $\overline{BD}$  gelöscht werden.

Die GN-Taste der Tastatur wird über PIN 70 und ein Eingangsfilter über WOR an PIN 46 auf das Verdrahtungsfeld zurückgegeben und mit der Netzteil-GN geodert.

### Ausgabeinformation



## CRT-INTERFACE MC 80-46

Der programmierbare Bildschirm-Controller BC 10 ist im Einfach-Europaformat aufgebaut und beinhaltet einen eigenen Bildwiederholtspeicher für eine Darstellung von max. 2048 Zeichen, die programmierbare Zeitsteuerung zur Erzeugung von Bild- und Zeilensynchronsignalen und die Schnittstellenelektronik zum Anschluß sowohl von TTL-kompatiblen Bildschirmen mit separaten Synchron- und Videoeingängen als auch von Monitoren mit Norm-BAS-Eingang nach RS 170. Die Adressierung der Zeichen erfolgt über einen 16 Bit breiten Adreßbus. Die Baugruppe belegt einen Adreßraum von 4 K Bytes, wobei die Basisadresse fest eingestellt ist. Die Zeichenadresse setzt sich aus der Zeilen- und Zeichenposition auf dem Schirm zusammen. Dieses X-Y-Verfahren ermöglicht eine einfache Positionierung von Zeichen und Cursor.

Das Schreiben und Lesen des Bildschirminhaltes und der Cursorposition, das Setzen der Parameter für Bild- und Rasterformat, und weitere Steuerfunktionen erfolgen über einen 8 Bit breiten bidirektionalen Datenbus. Die 128 Zeichen werden in einer 8 x 16 Punkte großen Matrix dargestellt, die auch Unterlängen erzeugt. Die Zeichen können sowohl horizontal wie vertikal nahtlos aneinandergereiht werden (Möglichkeit der graphischen Darstellung). Angesteuert werden die Daten im 7-Bit-ASCII-Code. Bei gesetztem achten Datenbit wird das Zeichen invers dargestellt. Durch eine zusätzliche Stapelkarte lassen sich die zeichenbegleitenden Sonderzeichen: Grau, Blinken, Grau-Blinken erzeugen. Das Bildschirmformat reicht bis 24 Zeilen zu 80 Zeichen; bei Beschränkung auf 64 Zeichen pro Zeile können bis zu 32 Zeilen dargestellt werden.

Für Bildformate bis 16 Zeilen x 64 Zeichen ist die Baugruppe mit einem 1 K Byte großen Bildwiederholtspeicher ausgestattet. Die programmierbare Zeitsteuerung ist mit dem Steuerbaustein CRT 5027 von SMC aufgebaut.

Für alle TTL-Bauteile werden ICs der Serie 74 LS eingesetzt. Die Baugruppe programmierbarer Bildschirm-Controller PBC-Karte benötigt eine Stromversorgung von + 5 V, + 12 V, - 12 V.



# CRT-INTERFACE MC 80-46

## PIN Belegungsliste

| PIN | Bezeichnung | Diagramm | Erläuterung    |
|-----|-------------|----------|----------------|
| 01  | + 5 V       |          |                |
| 02  | 0 V         |          |                |
| 03  | + 12 V      |          |                |
| 04  | 0 V         |          |                |
| 05  | - 12 V      |          |                |
| 06  | 0 V         |          |                |
| 07  |             |          |                |
| 08  |             |          |                |
| 09  | ADR 0       |          | Adresse-BUS 0  |
| 10  | ADR 1       |          | Adresse-BUS 1  |
| 11  | ADR 2       |          | Adresse-BUS 2  |
| 12  | ADR 3       |          | Adresse-BUS 3  |
| 13  | ADR 4       |          | Adresse-BUS 4  |
| 14  | ADR 5       |          | Adresse-BUS 5  |
| 15  | ADR 6       |          | Adresse-BUS 6  |
| 16  | ADR 7       |          | Adresse-BUS 7  |
| 17  | ADR 8       |          | Adresse-BUS 8  |
| 18  | ADR 9       |          | Adresse-BUS 9  |
| 19  | ADR 10      |          | Adresse-BUS 10 |
| 20  | ADR 11      |          | Adresse-BUS 11 |
| 21  | ADR 12      |          | Adresse-BUS 12 |
| 22  | ADR 13      |          | Adresse-BUS 13 |
| 23  | ADR 14      |          | Adresse-BUS 14 |
| 24  | ADR 15      |          | Adresse-BUS 15 |
| 25  | DAT 0       |          | Daten-BUS 0    |
| 26  | DAT 1       |          | Daten-BUS 1    |
| 27  | DAT 2       |          | Daten-BUS 2    |
| 28  | DAT 3       |          | Daten-BUS 3    |
| 29  | DAT 4       |          | Daten-BUS 4    |
| 30  | DAT 5       |          | Daten-BUS 5    |
| 31  | DAT 6       |          | Daten-BUS 6    |
| 32  | DAT 7       |          | Daten-BUS 7    |

| PIN | Bezeichnung | Diagramm | Erläuterung    |
|-----|-------------|----------|----------------|
| 33  | +5 VL       |          | +5V Logiksp.   |
| 34  | 0VL         |          | 0V Logikmasse  |
| 35  | +12VL       |          | +12V Logiksp.  |
| 36  | 0 VL        |          | 0 V Logikmasse |
| 37  | -12VL       |          | -12V Logiksp.  |
| 38  | 0 VL        |          | 0 V Logikmasse |
| 39  |             |          |                |
| 40  |             |          |                |
| 41  | MEMR        |          | Memory Read    |
| 42  | MEMW        |          | Memory Write   |
| 43  |             |          |                |
| 44  |             |          |                |
| 45  |             |          |                |
| 46  |             |          |                |
| 47  |             |          |                |
| 48  |             |          |                |
| 49  |             |          |                |
| 50  |             |          |                |
| 51  |             |          |                |
| 52  |             |          |                |
| 53  | PHI 2       |          | CPU-Takt 0 2   |
| 54  |             |          |                |
| 55  |             |          |                |
| 56  | INT         |          | Interrupt      |
| 57  | IRO         |          | Interrupt 0    |
| 58  | IR 1        |          | Interrupt 1    |
| 59  | IR 2        |          | Interrupt 2    |
| 60  | IR 3        |          | Interrupt 3    |
| 61  | IR 4        |          | Interrupt 4    |
| 62  | IR 5        |          | Interrupt 5    |
| 63  | IR 6        |          | Interrupt 6    |
| 64  | IR 7        |          | Interrupt 7    |

| PIN | Bezeichnung | Diagramm | Erläuterung     |
|-----|-------------|----------|-----------------|
| 65  |             |          |                 |
| 66  |             |          |                 |
| 67  |             |          |                 |
| 68  | BAS         |          | Bild-Austast-S. |
| 69  |             |          |                 |
| 70  |             |          |                 |
| 71  |             |          |                 |
| 72  |             |          |                 |
| 73  |             |          |                 |
| 74  |             |          |                 |
| 75  |             |          |                 |
| 76  |             |          |                 |
| 77  |             |          |                 |
| 78  |             |          |                 |
| 79  |             |          |                 |
| 80  |             |          |                 |
| 81  | HOR         |          | Zeilen-Impuls   |
| 82  | VERT        |          | Bild Impuls     |
| 83  |             |          |                 |
| 84  |             |          |                 |
| 85  |             |          |                 |
| 86  | VIDEO       |          | Video-Införm.   |
| 87  |             |          |                 |
| 88  |             |          |                 |
| 89  |             |          |                 |
| 90  |             |          |                 |
| 91  |             |          |                 |
| 92  |             |          |                 |
| 93  |             |          |                 |
| 94  |             |          |                 |
| 95  |             |          |                 |
| 96  |             |          |                 |





# CRT-INTERFACE MC 80-46

## Funktionsbeschreibung

Der CRT-Controller ist aufgeteilt in die Funktionsgruppen:

- Adreßerkennung und Adreßauswahl
- Adreßmodifizierung
  
- Programmierbare Zeitsteuerung
- Bildwiederholpeicher mit Ausgabelatch
- Zeichengenerator
- Videosignal Erzeugung
- BAS-Generator
- Taktgenerator

Der CRT-Controller verhält sich zur BUS-Schnittstelle wie ein 4 K-RAM-Speicher. Die oberen 4 Bits der auf 3 000 H liegenden Basisadresse aktiviert das Signal ADR.

Die restlichen 12 Adreßbits sind im Bildwiederholpeicher so aufgeteilt, daß jedes auf dem Bildschirm dargestellte Zeichen Teil eines x-y-Koordinatensystems ist. Dabei bilden die Adreßbits A0-A6 die Zeichenposition innerhalb der Zeile, die Adreßbits A7 - A11 die Zeilennummer.

Bei einem externen (Rechner-)Zugriff wird außer dem Signal ADR entweder MEMR oder MEMW aktiviert, und damit auch das Signal EXT. Dieses schaltet die 3 Adreßmultiplexer von interner Refreshadresse auf externe Rechneradresse um.

### **Adreßerkennung und Adreßauswahl**



# CRT-INTERFACE MC 80-46

## Funktionsbeschreibung

### Adreßmodifikation

Die über den BUS kommenden 12 Zeichenadreß-bits werden auf der Karte zu einer 11 Bit-Adresse reduziert, um so den Speicherplatz dem tatsächlich benötigten Bedarf anzupassen. Diese Reduzierung erfolgt durch einen weiteren Multiplexer.

Die folgende Darstellung zeigt die CRT-Adressenzuordnung für das Darstellungsformat *24 Zeilen x 80 Zeichen*. Der Adreßbereich für die darzustellenden Zeichen liegt zwischen 3000H und 3BCF H.

|       |  | Spalte |      |      |    |    |    |  |          |
|-------|--|--------|------|------|----|----|----|--|----------|
|       |  | 1      | 2    | .... | 78 | 79 | 80 |  |          |
| Zeile |  |        |      |      |    |    |    |  |          |
| 1     |  | 3000   | 3001 |      |    |    |    |  | ... 305F |
| 2     |  | 3080   |      |      |    |    |    |  |          |
| 3     |  | 3100   |      |      |    |    |    |  |          |
| 4     |  | 3180   |      |      |    |    |    |  |          |
| 5     |  | .      |      |      |    |    |    |  |          |
| 6     |  | .      |      |      |    |    |    |  |          |
| 7     |  | .      |      |      |    |    |    |  |          |
| 8     |  | .      |      |      |    |    |    |  |          |
| 9     |  | .      |      |      |    |    |    |  |          |
| 10    |  | .      |      |      |    |    |    |  |          |
| 11    |  | .      |      |      |    |    |    |  |          |
| 12    |  | .      |      |      |    |    |    |  |          |
| 13    |  | .      |      |      |    |    |    |  |          |
| 14    |  | .      |      |      |    |    |    |  |          |
| 15    |  | .      |      |      |    |    |    |  |          |
| 16    |  | .      |      |      |    |    |    |  |          |
| 17    |  | .      |      |      |    |    |    |  |          |
| 18    |  | .      |      |      |    |    |    |  |          |
| 19    |  | .      |      |      |    |    |    |  |          |
| 20    |  | .      |      |      |    |    |    |  |          |
| 21    |  | .      |      |      |    |    |    |  |          |
| 22    |  | .      |      |      |    |    |    |  |          |
| 23    |  | .      |      |      |    |    |    |  |          |
| 24    |  | 3B80   |      |      |    |    |    |  | 3BCF     |

- Bei den Steueradressen "Interrupt Request" und Zeitablaufsteuerung tritt kein Speicher-Chip-Select auf, so daß eine Doppelbeschreibung des Speichers vermieden wird.
- Der Speicherbereich für die Zeichen 65 - 80 in Zeile 1 - 24 und Zeichen 1 - 64 in Zeile 25-32 ist identisch, so daß die größtmöglichen Formate 32 Zeilen - 64 Zeichen bzw. 24 Zeilen - 80 Zeichen sind.

Der Interrupt-Request dient zur Synchronisierung des Rechnerzugriffes auf die vertikale Austastlücke, so daß der externe Zugriff während der Dunkelastung des Bildschirms erfolgt. Dadurch wird eine Störung des Bildes beim Rechnerzugriff vermieden.



## **CRT-INTERFACE MC 80-46**

### **Funktionsbeschreibung**

Die programmierbare Zeitsteuerung erfolgt durch den Controller Chip CRT 5027. Seine Steuerregister werden durch die unteren 4 Adreßbits adressiert und über den Datenbus geladen. Dazu wird auf der Karte, außer dem Chip Select auch ein Data-Strobe erzeugt. Der Controller Chip erzeugt dann über die Ausgänge  $H_0 - H_6$  (Zeichenposition) und  $DR_0 - DR_4$  (Zeilennummer) die interne Refresh-Adresse für den Bildwiederholtspeicher und über  $R_0 - R_3$  (Row Scan) die aktuelle Rasterzeilennummer für den Zeichengenerator.

Der Baustein erzeugt außerdem den horizontalen und vertikalen Synchronisationsimpuls, sowie ein Gemisch dieser Impulse für das BAS-Signal. Ferner liefert der Baustein das Cursor-Signal und das Blank-Signal für die Austastlücken.

### **Programmierbare Zeitsteuerung**

Die CRT-Controllerplatine benötigt je nach gewünschtem Darstellungsformat (16 x 32, 16 x 64, 24 x 80, Zeilen x Zeichen) einen 1 KB bzw. 2 KB großen Bildwiederholtspeicher, der durch die Multiplexer der Adressenauswahl adressiert wird. Die 8-Bit Datenausgänge dieses Speichers werden für einen Zeichenzyklus mit dem Signal DCC zwischengespeichert. Die unteren 7-Bit entsprechen dem ASCII-Code, während das 8. Bit zu einer zeichenbegleitenden Inversdarstellung benutzt wird.

### **Bildwiederholtspeicher**

Der ASCII-Code des Bildwiederholtspeichers bildet zusammen mit den 4 Bit ( $R_0 - R_3$ ) des Controller Chip die Adresse für den Zeichengenerator (2 K Byte Prom mit max. 128 Zeichen).

Mit dem Takt DCC werden die Daten dieses PROM'S parallel in ein Schieberegister übernommen und mit dem Takt DC in eine serielle Form gebracht.

### **Zeichengenerator**



alphatronic

## **CRT-INTERFACE MC 80-46**

### **Funktionsbeschreibung**

Die serielle Videoinformation des Schieberegisters wird mit dem aus Synchronisierungsgründen zwischengespeicherten Invers- bzw. Cursorbit verknüpft, das ebenso wie das Blanksignal (Horizontale vertikale Austastlücke) vom Controller Chip erzeugt wird.

Über ein WOR kann das Blank-Signal durch die Zusatzplatine modifiziert werden und somit die Grausteuerng bzw. das Blinken erzielt werden.

### **Video-Signal-Erzeugung**

---

Der BAS-Transistor-Generator liefert ein Signalgemisch niedriger Ausgangsimpedanz zur Ansteuerung von Monitoren mit BAS-Anschluß.

### **BAS-Generator**

---

Die Signale DC und DCC werden von einem Quarzgenerator bzw. von dem nachgeschalteten Synchronzähler geliefert. Dieser wird mit dem Videodot Clock getaktet und erzeugt bei Überlauf das Signal DCC. Mit der nächsten Taktflanke setzt er sich auf den Anfangszustand 8.

### **Taktgenerator**



## 16k RAM MC 80-10 RAM Speicher Platine

### Technische Beschreibung:

Bei der Baugruppe MC 80-10 handelt es sich um einen statischen Schreib/Lese-Speicher mit einer Gesamtkapazität von 16 K-Byte. Bestückt ist diese Karte mit 4 K-Bit RAMs vom Typ 2114L (Intel) bzw. TMS 4045-45NL (Texas Instruments).

Die maximale Zugriffszeit beträgt 450 nsec.

Bei maximaler Bestückung befinden sich 32 RAMs auf einer Speicherkarte. Eine Teilbestückung kann in 4K-Schritten erfolgen.

Alle Adress-, Daten- und Steuerleitungen sind voll gepuffert.

Durch einen Adresswahlschalter auf der Platine kann der Speicherbereich auf eine der 4 16K-Blöcke des Gesamtadressbereiches gelegt werden.

Stromversorgung: +5V  $\pm$  5% typ.: 2,1A max.: 2,5 A

Der interne Adreß-Bus wird mit den Invertern C 1 und D 1 (SN 74 LS 04) vom MC 80-Adreß-Bus entkoppelt. Die Entkoppelung zwischen dem internen Daten-Bus und dem MC 80-Daten-Bus erfolgt mit den invertierenden bidirektionalen Bus-Treibern 8226 (INTEL) F 1, G 1.

### **Funktions-Beschreibung ADRESS/DATEN-BUS**

Die Adreßerkennung erfolgt mittels eines Schaltnetzes, wobei die Adressenbits  $\overline{ADR 14}$  und  $\overline{ADR 15}$  mit den Schalterstellungen 3 und 4 verglichen werden. Bei Gleichheit werden die zwei "1 aus 8-Dekoder" 8205 (INTEL) freigegeben, die in Anhängigkeit der Adressenbits  $\overline{ADR 10}$  bis  $\overline{ADR 13}$  die Chipselektimpulse  $\overline{CS 0}$  -  $\overline{CS 15}$  erzeugen.

### **ADRESSDEKODIERUNG**



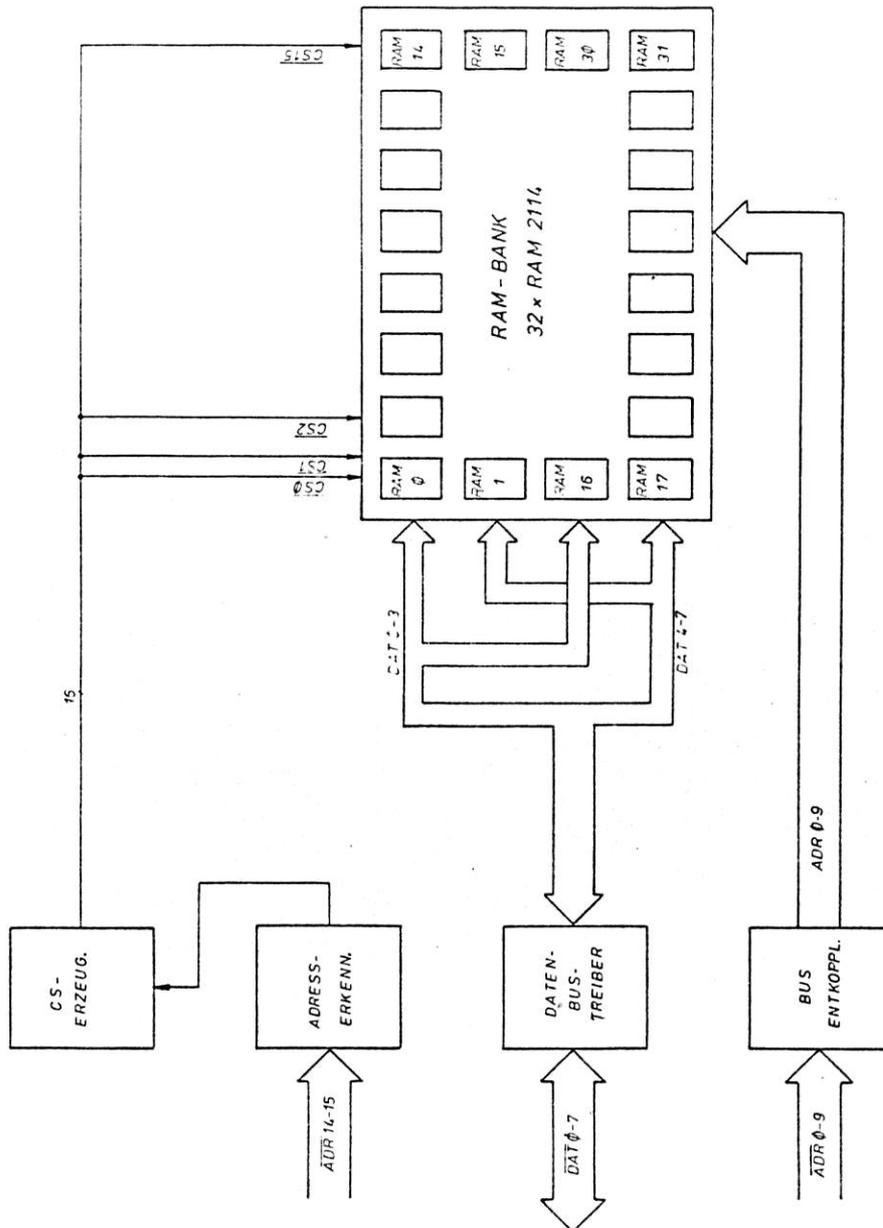
alphatronic

# 16k RAM MC 80-10

## PIN Belegung

| PIN | Bezeichnung | Diagramm | Erläuterung    |
|-----|-------------|----------|----------------|
| 01  | + 5 V       |          |                |
| 02  | 0 V         |          |                |
| 03  |             |          |                |
| 04  | 0 V         |          |                |
| 05  |             |          |                |
| 06  | 0 V         |          |                |
| 07  |             |          |                |
| 08  |             |          |                |
| 09  | ADR0        |          | Adresse-BUS 0  |
| 10  | ADR1        |          | Adresse-BUS 1  |
| 11  | ADR2        |          | Adresse-BUS 2  |
| 12  | ADR3        |          | Adresse-BUS 3  |
| 13  | ADR4        |          | Adresse-BUS 4  |
| 14  | ADR5        |          | Adresse-BUS 5  |
| 15  | ADR6        |          | Adresse-BUS 6  |
| 16  | ADR7        |          | Adresse-BUS 7  |
| 17  | ADR8        |          | Adresse-BUS 8  |
| 18  | ADR9        |          | Adresse-BUS 9  |
| 19  | ADR10       |          | Adresse-BUS 10 |
| 20  | ADR11       |          | Adresse-BUS 11 |
| 21  | ADR12       |          | Adresse-BUS 12 |
| 22  | ADR13       |          | Adresse-BUS 13 |
| 23  | ADR14       |          | Adresse-BUS 14 |
| 24  | ADR15       |          | Adresse-BUS 15 |
| 25  | DAT0        |          | Daten-BUS 0    |
| 26  | DAT1        |          | Daten-BUS 1    |
| 27  | DAT2        |          | Daten-BUS 2    |
| 28  | DAT3        |          | Daten-BUS 3    |
| 29  | DAT4        |          | Daten-BUS 4    |
| 30  | DAT5        |          | Daten-BUS 5    |
| 31  | DAT6        |          | Daten-BUS 6    |
| 32  | DAT7        |          | Daten-BUS 7    |

| PIN | Bezeichnung | Diagramm | Erläuterung  |
|-----|-------------|----------|--------------|
| 33  | + 5 V       |          |              |
| 34  | 0 V         |          |              |
| 35  |             |          |              |
| 36  | 0 V         |          |              |
| 37  |             |          |              |
| 38  | 0 V         |          |              |
| 39  |             |          |              |
| 40  |             |          |              |
| 41  | MEMR        |          | Memory Read  |
| 42  | MEMW        |          | Memory Write |
| 43  |             |          |              |
| 44  |             |          |              |
| 45  |             |          |              |
| 46  |             |          |              |
| 47  |             |          |              |
| 48  |             |          |              |
| 49  |             |          |              |
| 50  |             |          |              |
| 51  |             |          |              |
| 52  |             |          |              |
| 53  |             |          |              |
| 54  |             |          |              |
| 55  |             |          |              |
| 56  |             |          |              |
| 57  |             |          |              |
| 58  |             |          |              |
| 59  |             |          |              |
| 60  |             |          |              |
| 61  |             |          |              |
| 62  |             |          |              |
| 63  |             |          |              |
| 64  |             |          |              |





## Primär-getaktetes Netzgerät Grundmodul g1

### SPEZIFIKATIONEN

|                                            |                                                                                          |
|--------------------------------------------|------------------------------------------------------------------------------------------|
| Eingangsspannung (AC)                      | 187 V bis 242 V                                                                          |
| Eingangsfrequenz                           | 50 Hz / 60 Hz                                                                            |
| Ausgangsspannung (DC)                      | 5 V                                                                                      |
| Einstellbereich der Ausgangsspannung       | 4,4 V bis 5,6 V                                                                          |
| Ausgangsstrom                              | 1 A bis 10 A                                                                             |
| Restwelligkeit bei Vollast                 | $\leq 100$ mVss                                                                          |
| Überlastschutz                             | Abschaltung bei Kurzschluß                                                               |
| Einstellbereich der Strombegrenzung        | 6,5 A bis 11 A                                                                           |
| Überspannungsschutz                        | Gerät schaltet ab                                                                        |
| Ansprechschwelle des Überspannungsschutzes | 5,7 V bis 6 V                                                                            |
| Ansprechzeit des Überspannungsschutzes     | $\leq 1$ ms                                                                              |
| Regelverhalten der Ausgangsspannung        | max. 2,5% bei Laständerung von 0-100% u. gleichzeitiger Netzschwankungen von +10% / -15% |
| Ausregelzeit                               | 1,5 ms bei Lastsprung von 0 - 100%                                                       |
| Umgebungstemperatur                        | 0°C bis 40°C                                                                             |
| Belüftung                                  | $\geq 50$ m <sup>3</sup> / h                                                             |
| Wirkungsgrad bei Vollast                   | $\geq 75\%$                                                                              |



# STROMVERSORGUNG

## Längsreguliertes Netzgerät Erweiterungsmodul ei

### SPEZIFIKATIONEN (für eine Regelstrecke)

Das Modul ei enthält 3 12V-Regelstrecken

|                                      |                                                   |
|--------------------------------------|---------------------------------------------------|
| Eingangsspannung (AC > 30 Hz)        | 17 V bis 35 V                                     |
| Eingangsspannung (DC)                | 15 V bis 35 V                                     |
| Ausgangsspannung (DC)                | 12 V                                              |
| Einstellbereich der Ausgangsspannung | 11,5 V bis 12,5 V                                 |
| Ausgangsstrom                        | 2 A (4A für $t \leq 40\text{ms}$ )                |
| Restwelligkeit bei 2 A               | $\leq 120 \text{ mVss}$                           |
| Überlastschutz                       | Interne Strombegrenzung und thermische Sicherung. |
| Lastregelfaktor                      | $\leq 1\%$                                        |
| Netzregelfaktor                      | $\leq 1\%$                                        |
| Umgebungstemperatur                  | 0°C bis 40°C                                      |
| Belüftung                            | $\geq 50 \text{ m}^3/\text{h}$                    |

Eingebaute Logik

Datenschutzsignal, General Null und Uhrentakt.



alphaTronic

## MINI FLOPPY IF MC 80-60

Mini Floppy Interface

Die Baugruppe "Mini-Floppy Basis Controller" MC 80 - 60 ist eine Grundelektronik, die zur Datenübertragung zwischen Rechner und Mini-Floppy-Speicher dient. Mini-Floppy-Laufwerke (maximal zwei) vom Typ BASF 6106 oder ähnliche Geräte können über Basis Controller am MC 80-BUS oder anderen BUS-orientierten Rechnersystemen betrieben werden. Beim Datentransfer zwischen System-BUS und Mini-Floppy hat der Basis Controller im wesentlichen die folgenden Aufgaben:

- BUS - Entkopplung
- Zwischenpufferung der Geräteadresse und der Steuersignale
- Zeichenweise Datenpufferung
- Datendekodierung
- Datenformatierung
- Schreibtaktzeugung

Der Basis Controller ist unter Verwendung des Floppy Disk Formatter/ Controller - Bausteins FD 1771-1 von Western Digital realisiert worden.

### Allgemeines

---

Die Schnittstelle zwischen Basis Controller und MC - 80 - BUS besteht aus dem bidirektionalen 8-Bit-DATEN-BUS, dem 8-Bit-ADRESS-BUS und dem 5-Bit-CONTROL-BUS. Alle BUS-Signale sind über Treiber/ Empfänger in Tri-State-Technik vom System-BUS entkoppelt.

### Funktionsgruppenübersicht



# MINI FLOPPY IF MC 80-60

## PIN Belegung

| PIN | Bezeichnung | Diagramm | Erläuterung   | PIN | Bezeichnung | Diagramm | Erläuterung        | PIN | Bezeichnung | Diagramm | Erläuterung                        |
|-----|-------------|----------|---------------|-----|-------------|----------|--------------------|-----|-------------|----------|------------------------------------|
| 01  | + 5V        |          |               | 33  | +5 VL       |          | +5V Logiksp.       | 65  | MOTOR ON    |          | Motor-Start-Signal                 |
| 02  | 0V          |          |               | 34  | 0VL         |          | 0V Logikmasse      | 66  | SIDE SEL    |          | Diskettenseitenanwahl              |
| 03  | +12V        |          |               | 35  | +12 VL      |          | + 12 V Logiksp.    | 67  | WRITE GATE  |          | Schreibstromfreigabe               |
| 04  | 0V          |          |               | 36  | 0 VL        |          | 0 V Logikmasse     | 68  | WRITE DATA  |          | Schreibdaten                       |
| 05  | -12V        |          |               | 37  | -12 VL      |          | -12V Logiksp.      | 69  | DIR IN      |          | Richtung zur Diskettenmitte        |
| 06  | 0V          |          |               | 38  | 0 VL        |          | 0 V Logikmasse     |     |             |          |                                    |
| 07  |             |          |               | 39  |             |          |                    | 70  | WRITE PROT. |          | Schreibschutz an                   |
| 08  |             |          |               | 40  |             |          |                    | 71  | READY       |          | Antrieb-Bereit-Meldg.              |
| 09  | ADR 0       |          | Adresse-BUS 0 | 41  |             |          |                    | 72  | READ DATA   |          | Ungetrennte Lesedaten              |
| 10  | ADR 1       |          | Adresse-BUS 1 | 42  |             |          |                    | 73  | SPARE       |          | Reserve-Leitung                    |
| 11  | ADR 2       |          | Adresse-BUS 2 | 43  | IOR         |          | Input-Output-Read  | 74  |             |          |                                    |
| 12  | ADR 3       |          | Adresse-BUS 3 | 44  | IOW         |          | Input-Output-Write | 75  |             |          |                                    |
| 13  | ADR 4       |          | Adresse-BUS 4 | 45  |             |          |                    | 76  | TRACK 00    |          | Spur-Null-Position                 |
| 14  | ADR 5       |          | Adresse-BUS 5 | 46  |             |          |                    | 77  |             |          |                                    |
| 15  | ADR 6       |          | Adresse-BUS 6 | 47  | RST         |          | Reset intern       | 78  |             |          |                                    |
| 16  | ADR 7       |          | Adresse-BUS 7 | 48  | RDY         |          | Ready              | 79  |             |          |                                    |
| 17  |             |          |               | 49  | WAIT        |          | Wait               | 81  |             |          |                                    |
| 18  |             |          |               | 50  |             |          |                    | 81  | SEL 1       |          | Anwahl Antrieb 1                   |
| 19  |             |          |               | 51  |             |          |                    | 82  | SEL 2       |          | Anwahl Antrieb 2                   |
| 20  |             |          |               | 52  |             |          |                    | 83  | STEP        |          | Schrittmotorpuls                   |
| 21  |             |          |               | 53  |             |          |                    | 84  | HEAD LOAD   |          | Schreib-Lese-Kopfanziehen          |
| 22  |             |          |               | 54  |             |          |                    | 85  | SEL 3       |          | Anwahl Antrieb 3                   |
| 23  |             |          |               | 55  |             |          |                    | 86  | SOFTWINTL   |          | Softwaremäßige Klappenverriegelung |
| 24  |             |          |               | 56  | INT         |          | Interrupt          | 87  |             |          |                                    |
| 25  | DAT 0       |          | Daten-BUS 0   | 57  | IRO         |          | Interrupt 0        | 88  |             |          |                                    |
| 26  | DAT 1       |          | Daten-BUS 1   | 58  | IR 1        |          | Interrupt 1        | 89  |             |          |                                    |
| 27  | DAT 2       |          | Daten-BUS 2   | 59  | IR 2        |          | Interrupt 2        | 90  |             |          |                                    |
| 28  | DAT 3       |          | Daten-BUS 3   | 60  | IR 3        |          | Interrupt 3        | 91  |             |          |                                    |
| 29  | DAT 4       |          | Daten-BUS 4   | 61  | IR 4        |          | Interrupt 4        | 92  | INDEX       |          | Index Puls                         |
| 30  | DAT 5       |          | Daten-BUS 5   | 62  | IR 5        |          | Interrupt 5        | 93  |             |          |                                    |
| 31  | DAT 6       |          | Daten-BUS 6   | 63  | IR 6        |          | Interrupt 6        | 94  |             |          |                                    |
| 32  | DAT 7       |          | Daten-BUS 7   | 64  | IR 7        |          | Interrupt 7        | 95  | 0 Volt      |          |                                    |
|     |             |          |               |     |             |          |                    | 96  |             |          |                                    |



alphaTronic

## PERIPHERIE

### Mini-Floppy-Drive

Der Mini-Floppy-Drive ist ein kleiner, kompakter, zuverlässiger und preiswerter Floppy-Disk-Speicher. Der MFD ist als peripherer Datenspeicher für Klein- und Tischrechner, Kassenterminale und Textverarbeitung konzipiert.

In vielen Anwendungsfällen bietet der Mini-Floppy-Drive eine interessante Alternative bei vergleichbarer Größe zu Lochstreifen-Ein/Ausgabe-Geräten, Magnetbandkassetten und Magnetkarten-Laufwerken.

Die Speicherkapazität beträgt 80 K Bytes je Laufwerk bei einer mittleren Zugriffszeit von 140 ms. Durch den mit der Drehung des Datenträgers (Diskette) lückenlos schritthaltenden Datentransfer wird eine mittlere Datenrate von 125 K Bits/Sek. ermöglicht.

Jedes Laufwerk enthält eine auswechselbare Diskette. Die Positioniereinrichtung führt den Schreib-Lese-Kopf zu der gewünschten Spur, und ein Kopfaktivierungsmechanismus drückt die Diskette nur beim Lese- und Schreibvorgang an den Schreib-Lese-Kopf.

Als peripherer Speicher mit wahlfreiem Zugriff ermöglicht es die Floppy Disk, ein einzelnes Byte als kleinstes adressierbares Datenelement auf dem Datenträger anzusprechen. An den Computer sind bis zu zwei Floppy-Disk-Laufwerke anschließbar.

Die Diskette ist eine einseitig magnetbeschichtete flexible Kunststoffplatte, die in eine fest verschlossene Schutzhülle drehbar eingebettet ist. Die Informationsspeicherung erfolgt in Wechseltaktschrift auf 40 konzentrischen Spuren.



alphaTronic

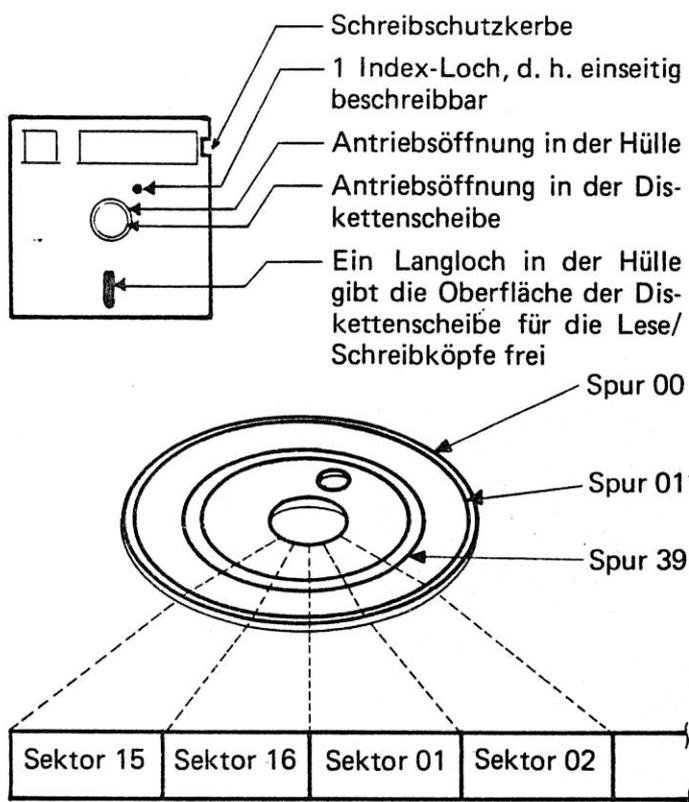
# PERIPHERIE

## Mini-Floppy-Drive

### TECHNISCHE DATEN

|                                                   |                                                                                        |
|---------------------------------------------------|----------------------------------------------------------------------------------------|
| ○ Anzahl Laufwerke                                | max. 2                                                                                 |
| ○ Datenträger                                     | 1 Diskette je Laufwerk                                                                 |
| Kapazität je Laufwerk                             | 80 K Bytes                                                                             |
| Datenoberflächen                                  | 1                                                                                      |
| Anzahl der Spuren                                 | 40                                                                                     |
| ○ Datenformat                                     | soft-sektoriert                                                                        |
| je Spur                                           | 16 Sektoren                                                                            |
| je Sektor                                         | 128 Bytes                                                                              |
| ○ Datenaufzeichnung                               | FM Wechseltaktschrift                                                                  |
| Bitdichte                                         | 2768 bpi                                                                               |
| Spurdichte                                        | 48 tpi                                                                                 |
| ○ Spindeldrehzahl                                 | 300 Umdrehungen/Min.                                                                   |
| ○ Positionierzeit                                 | 12 ms                                                                                  |
| ○ Mittelwert                                      | ----                                                                                   |
| Spur zu Spur                                      | 12 ms                                                                                  |
| ○ Mittlere Zugriffszeit                           | 140 ms                                                                                 |
| ○ Datenübertragungsrate                           | 125 K Bits/Sek.                                                                        |
| ○ Verfügbarkeit                                   |                                                                                        |
| MTBF                                              | 5 000 Betriebsstunden<br>bei starker Benutzung                                         |
|                                                   | 8 000 Betriebsstunden<br>bei normaler Benutzung                                        |
| MTRR                                              | 30 Minuten                                                                             |
| Fehlerraten                                       |                                                                                        |
| Lesefehler korrigierbar<br>(soft errors)          | 1 pro 10 <sup>8</sup> Bits                                                             |
| Lesefehler nicht korrigierbar<br>(hard errors)    | 1 pro 10 <sup>11</sup> Bits                                                            |
| Positionierfehler                                 | 1 pro 10 <sup>6</sup> Suchläufe                                                        |
| ○ Anschlußwerte:                                  | + 12 V, 0,65 A max.*<br>+ 5 V, 0,5 A max.<br>*Motoranlaufstrom<br>+ 1,2 A für 0,5 Sek. |
| ○ Leistungsaufnahme                               |                                                                                        |
| Gerät in Betrieb                                  | 10,5 Watt                                                                              |
| stand by (Motor aus)                              | 4,5 Watt                                                                               |
| Motor an und Gerät<br>nicht angewählt             | 8,0 Watt                                                                               |
| ○ Umgebungsbedingungen                            |                                                                                        |
| Temperatur                                        | 10 <sup>0</sup> C bis 50 <sup>0</sup> C                                                |
| rel. Luftfeuchte                                  | 20% bis 80%                                                                            |
| ○ Abmessungen                                     |                                                                                        |
| Breite                                            | 146,1 mm                                                                               |
| Höhe                                              | 43,5 mm                                                                                |
| Tiefe                                             | 190,0 mm                                                                               |
| ○ Gewicht                                         | 1,4 kg                                                                                 |
| ○ Diskette, BASF 606 Mini-Disk od. Vergleichbares |                                                                                        |
| Hülle                                             | 133,4 x 133,4 mm                                                                       |
| Außendurchmesser                                  | 130,2 mm                                                                               |
| Innendurchmesser                                  | 28,575 mm                                                                              |
| ○ Lebensdauer                                     |                                                                                        |
| Ladevorgänge                                      | 30 000                                                                                 |
| Durchläufe pro Spur                               | 5 x 10 <sup>6</sup>                                                                    |

Die Disketten für Ihren Computer werden einseitig beschrieben und sind soft-sektoriert. Soft-sektoriert heißt, daß die Sektorkennzeichnung auf der Diskette elektronisch aufgezeichnet ist im Gegensatz zur Festsektorierung, bei der die Sektoren durch Löcher in der Diskette gekennzeichnet sind. Die Einteilung und damit die Sektorkennzeichnung wird beim Computer vom Formatierungsprogramm vorweggenommen (sh. MOS-Kommandos.)



Die Speicherkapazität jeder Diskette setzt sich wie folgt zusammen:

| Begriff                    | Wert  | Einheit  |
|----------------------------|-------|----------|
| Anz. Spuren (Tracks)       | 40    | Spuren   |
| Anz. Sektoren pro Spur     | 16    | Sektoren |
| Anz. Sektoren pro Diskette | 640   | Sektoren |
| Anz. Bytes pro Sektor      | 128   | Bytes    |
| Anz. Bytes pro Spur        | 2048  | Bytes    |
| Anz. Bytes pro Diskette    | 81920 | Bytes    |

Für die Adressierung der Diskette ist zu beachten, daß die Benennung der Spuren von 00 bis 39 erfolgt im Gegensatz dazu werden die Sektoradressen von 01 bis 16 je Spur gegeben.

Weitere Leistungsdaten zum Thema Disketten und Diskettenlaufwerke entnehmen Sie bitte dem entsprechenden Kapitel.



alphatronic

## Disketten-Handling

Die Diskette ist eine kleine, flexible (biegsame) Magnetplatte, die immer in einer Schutzhülle steckt.

Beschädigte Disketten dürfen nicht benutzt werden, da dies zu Fehlern bei der Verarbeitung der Daten führen kann.

Wenn Sie die Diskette aus dem Umschlag nehmen, halten Sie sie an der oberen Kante und ziehen die Diskette heraus. Berühren Sie nicht die Oberfläche der in der Schutzhülle steckenden Platte. Sollten Sie die Diskette nicht mehr benötigen oder beschriften, stecken Sie sie in den Umschlag zurück.

Beschriften Sie den Aufkleber der Diskette nicht mit einem harten Bleistift oder Kugelschreiber, sondern am besten mit einem weichen Filzstift.

Benutzen Sie keine Büroklammern.

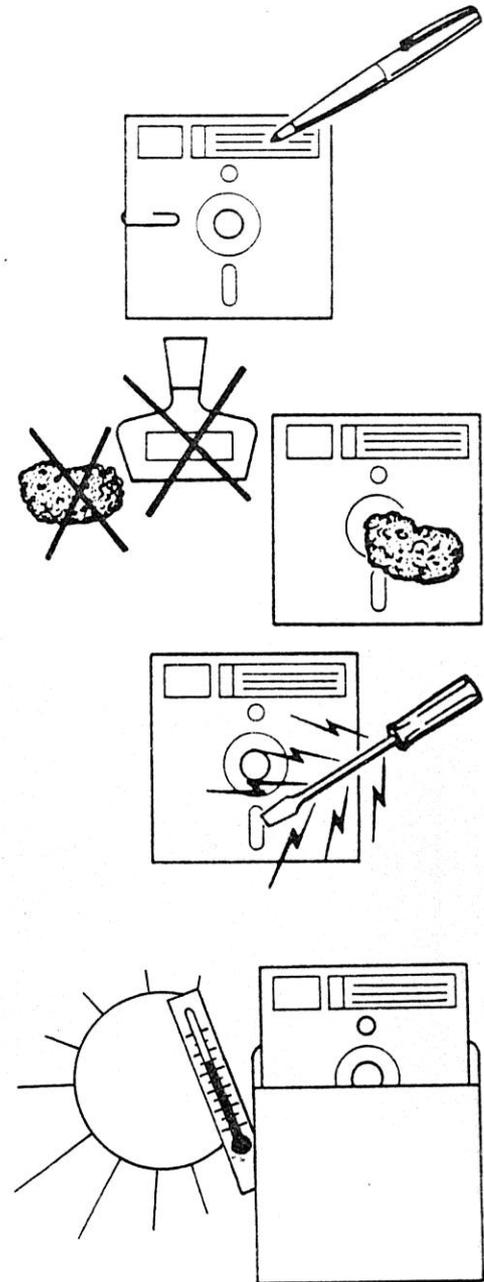
Reinigen Sie nicht die Oberfläche der Diskette.

Schützen Sie Ihre Disketten vor Flüssigkeiten, Speiseresten und dergleichen.

Stellen Sie keine schweren Gegenstände auf die Disketten.

Halten Sie die Diskette von Magneten fern, denn jede Diskette, die einem magnetischen Feld ausgesetzt wird, verliert Informationen.

Setzen Sie die Diskette keiner starken Hitze oder Sonnenbestrahlung aus.





# SCHREIBKERNDRUCKER SD 4035

Der Schreibkerndrucker SD 4035 zeichnet sich durch seine modulare Bauweise, den geringen Wartungsaufwand und besonders durch seine hohe Schriftqualität aus.

Detaillierte Angaben finden Sie in der separaten "Bedienungsanleitung für den Schreibkerndrucker SD 4035"

sowie in der "Spezifikation, Schreibkerndrucker mit Interface, Modell SD 4035"

|                                  |                                                                                                                                                    |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| Druckertyp                       | Schreibkerndrucker, serieller Druck<br>Schriftqualität wie elektrische<br>Schreibmaschine                                                          |
| Schreibkerne                     | 96 Zeichen; Groß- und Kleinbuch-<br>staben, OCR- und Proportional-<br>schriften                                                                    |
| Druckgeschwin-<br>digkeit        | 20 Zeichen/sek., Druckwegopti-<br>mierung, sowie Vor- und Rück-<br>wärtsdruck                                                                      |
| Teilung                          | 1/10, 1/12 und 1/15 Zoll,<br>kleinste Teilung 1/60 Zoll                                                                                            |
| Druckbreite                      | 132, 158 und 198 Zeichen/Zeile<br>je nach Schreibkern                                                                                              |
| Formularbreite                   | 375 mm                                                                                                                                             |
| Nutzen                           | 6 (1 Original, 5 Kopien)                                                                                                                           |
| Zeilenvorschub                   | 1/6 Zoll in 2 Richtungen                                                                                                                           |
| Plotten                          | kleinster Schritt 1/60 Zoll,<br>horizontal und vertikal                                                                                            |
| Leistungs-<br>elektronik         | eingebaute Ansteuereinheit für<br>Schrittmotore und Magnete                                                                                        |
| Stromversorgung                  | eingebautes Netzteil CAC 02 für<br>alle Betriebsspannungen, umschalt-<br>bar für Netzspannungen 110 V,<br>130 V, 220 V, 240 V Wechsel-<br>spannung |
| Schnittstellen -<br>seriell      | CCITT - V.24/V.28, EIA RS<br>232, TTY                                                                                                              |
| Schnittstellen -<br>parallel     | ASCII 8-Bit, Centronics, DATA<br>100, Dataproducts, Diablo, Qume                                                                                   |
| Übertragungsge-<br>schwindigkeit | 110 bis 9600 Baud                                                                                                                                  |
| Drucker mit Gehäuse:             |                                                                                                                                                    |
| Abmessungen                      | Breite 530 mm<br>Höhe 210 mm<br>Tiefe 300 mm                                                                                                       |
| Gewicht                          | ca. 23 kg                                                                                                                                          |
| Formular-<br>techniken           | – EFS 8, Endlosformular-Einrich-<br>tung-Simplex<br>Antrieb über Zeilenvorschub                                                                    |

## TECHNISCHE DATEN



alphaTronic

# ASCII TABELLE

| Dec Val | Hex Val | ASCII Code | ASCII Character |
|---------|---------|------------|-----------------|
| 000     | 00      | 0000000    | NUL             |
| 001     | 01      | 0000001    | SOH             |
| 002     | 02      | 0000010    | STX             |
| 003     | 03      | 0000011    | ETX             |
| 004     | 04      | 0000100    | EOT             |
| 005     | 05      | 0000101    | ENQ             |
| 006     | 06      | 0000110    | ACK             |
| 007     | 07      | 0000111    | BEL             |
| 008     | 08      | 0001000    | BS              |
| 009     | 09      | 0001001    | HT              |
| 010     | 0A      | 0001010    | LF              |
| 011     | 0B      | 0001011    | VT              |
| 012     | 0C      | 0001100    | FF              |
| 013     | 0D      | 0001101    | CR              |
| 014     | 0E      | 0001110    | SO              |
| 015     | 0F      | 0001111    | SI              |
| 016     | 10      | 0010000    | DLE             |
| 017     | 11      | 0010001    | DC1             |
| 018     | 12      | 0010010    | DC2             |
| 019     | 13      | 0010011    | DC3             |
| 020     | 14      | 0010100    | DC4             |
| 021     | 15      | 0010101    | NAK             |
| 022     | 16      | 0010110    | SYN             |
| 023     | 17      | 0010111    | ETB             |
| 024     | 18      | 0011000    | CAN             |
| 025     | 19      | 0011001    | EM              |
| 026     | 1A      | 0011010    | SUB             |
| 027     | 1B      | 0011011    | ESC             |
| 028     | 1C      | 0011100    | FS              |
| 029     | 1D      | 0011101    | GS              |
| 030     | 1E      | 0011110    | RS              |
| 031     | 1F      | 0011111    | US              |
| 032     | 20      | 0100000    | SPACE           |
| 033     | 21      | 0100001    | !               |
| 034     | 22      | 0100010    | "               |
| 035     | 23      | 0100011    | #               |
| 036     | 24      | 0100100    | \$              |
| 037     | 25      | 0100101    | %               |
| 038     | 26      | 0100110    | &               |
| 039     | 27      | 0100111    | '               |
| 040     | 28      | 0101000    | (               |
| 041     | 29      | 0101001    | )               |
| 042     | 2A      | 0101010    | *               |
| 043     | 2B      | 0101011    | +               |
| 044     | 2C      | 0101100    | ,               |
| 045     | 2D      | 0101101    | .               |
| 046     | 2E      | 0101110    | :               |
| 047     | 2F      | 0101111    | /               |

| Dec Val | Hex Val | ASCII Code | ASCII Character |
|---------|---------|------------|-----------------|
| 048     | 30      | 0110000    | 0               |
| 049     | 31      | 0110001    | 1               |
| 050     | 32      | 0110010    | 2               |
| 051     | 33      | 0110011    | 3               |
| 052     | 34      | 0110100    | 4               |
| 053     | 35      | 0110101    | 5               |
| 054     | 36      | 0110110    | 6               |
| 055     | 37      | 0110111    | 7               |
| 056     | 38      | 0111000    | 8               |
| 057     | 39      | 0111001    | 9               |
| 058     | 3A      | 0111010    | :               |
| 059     | 3B      | 0111011    | ;               |
| 060     | 3C      | 0111100    | <               |
| 061     | 3D      | 0111101    | =               |
| 062     | 3E      | 0111110    | >               |
| 063     | 3F      | 0111111    | ?               |
| 064     | 40      | 1000000    | @               |
| 065     | 41      | 1000001    | A               |
| 066     | 42      | 1000010    | B               |
| 067     | 43      | 1000011    | C               |
| 068     | 44      | 1000100    | D               |
| 069     | 45      | 1000101    | E               |
| 070     | 46      | 1000110    | F               |
| 071     | 47      | 1000111    | G               |
| 072     | 48      | 1001000    | H               |
| 073     | 49      | 1001001    | I               |
| 074     | 4A      | 1001010    | J               |
| 075     | 4B      | 1001011    | K               |
| 076     | 4C      | 1001100    | L               |
| 077     | 4D      | 1001101    | M               |
| 078     | 4E      | 1001110    | N               |
| 079     | 4F      | 1001111    | O               |
| 080     | 50      | 1010000    | P               |
| 081     | 51      | 1010001    | Q               |
| 082     | 52      | 1010010    | R               |
| 083     | 53      | 1010011    | S               |
| 084     | 54      | 1010100    | T               |
| 085     | 55      | 1010101    | U               |
| 086     | 56      | 1010110    | V               |
| 087     | 57      | 1010111    | W               |
| 088     | 58      | 1011000    | X               |
| 089     | 59      | 1011001    | Y               |
| 090     | 5A      | 1011010    | Z               |
| 091     | 5B      | 1011011    | [               |
| 092     | 5C      | 1011100    | \               |
| 093     | 5D      | 1011101    | ]               |
| 094     | 5E      | 1011110    | ^               |
| 095     | 5F      | 1011111    | _               |

| Dec Val | Hex Val | ASCII Code | ASCII Character |
|---------|---------|------------|-----------------|
| 096     | 60      | 1100000    |                 |
| 097     | 61      | 1100001    | a               |
| 098     | 62      | 1100010    | b               |
| 099     | 63      | 1100011    | c               |
| 100     | 64      | 1100100    | d               |
| 101     | 65      | 1100101    | e               |
| 102     | 66      | 1100110    | f               |
| 103     | 67      | 1100111    | g               |
| 104     | 68      | 1101000    | h               |
| 105     | 69      | 1101001    | i               |
| 106     | 6A      | 1101010    | j               |
| 107     | 6B      | 1101011    | k               |
| 108     | 6C      | 1101100    | l               |
| 109     | 6D      | 1101101    | m               |
| 110     | 6E      | 1101110    | n               |
| 111     | 6F      | 1101111    | o               |
| 112     | 70      | 1110000    | p               |
| 113     | 71      | 1110001    | q               |
| 114     | 72      | 1110010    | r               |
| 115     | 73      | 1110011    | s               |
| 116     | 74      | 1110100    | t               |
| 117     | 75      | 1110101    | u               |
| 118     | 76      | 1110110    | v               |
| 119     | 77      | 1110111    | w               |
| 120     | 78      | 1111000    | x               |
| 121     | 79      | 1111001    | y               |
| 122     | 7A      | 1111010    | z               |
| 123     | 7B      | 1111011    | {               |
| 124     | 7C      | 1111100    |                 |
| 125     | 7D      | 1111101    | }               |
| 126     | 7E      | 1111110    | ~               |
| 127     | 7F      | 1111111    | DEL             |



# BASIC COMPILER

## Fehlermeldungen und Fehlercodes

### Warnungen

|                                              |   |
|----------------------------------------------|---|
| Auf angegebene Variable erfolgt kein Zugriff | 1 |
| Keine Zuweisung an angegebene Variable       | 2 |

Beispiel

```
C$ WARNING 2
ISA  xxxxH xxxxH
VSA  xxxxH xxxxH
```

Diese Fehler-Nummern können im BASIC-Programm bei Verwendung der Variablen ERR direkt abgefragt werden.

### Fehlermeldungen der Library

Beispiel

```
10 IF ERR = 5.0 THEN 20
20 PRINT ,ERR=, ERR
```

### BEDEUTUNG

### Fehler-Nr.

|                                                       |    |
|-------------------------------------------------------|----|
| Überlauf                                              | 1  |
| Längenfehler (nur BCD Strings)                        | 2  |
| Divisor ist Null                                      | 3  |
| Operand ist unnormalisierte Gleitkommazahl            | 6  |
| Falsche Eingabe im Rahmen des INPUT-Statements        | 8  |
| Negatives Argument bei logarithmischen Funktionen     | 10 |
| Argument bei logarithmischen Funktionen ist Null      | 11 |
| Argument bei trigonometrischen Funktionen ist zu groß | 12 |
| Überlauf Index                                        | 13 |
| Datenfehler                                           | 30 |
| Gerät nicht bereit (allgem. Fehler)                   | 31 |
| Datenträgerende                                       | 32 |
| Dateiende (Filemark)                                  | 33 |

Beispiel:

```
I/O 33
```



alphaTronic

# BASIC COMPILER

Fehlermeldungen und Fehlercodes

## Fehlercodes der Analysephase

### BEDEUTUNG

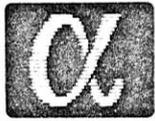
### CODE

|                                               |    |
|-----------------------------------------------|----|
| Zeilennummer fehlt                            | 01 |
| Zeilennummer zu groß                          | 02 |
| Zeile endet nach der Zeilennummer             | 03 |
| BASIC-Anweisung fehlt oder ist unzulässig     | 04 |
| Zeile ist zu lang (mehr als 69 Zeichen)       | 05 |
| Zeile enthält zu viele Fehler                 | 06 |
| Zeilenende vorzeitig gefunden (unvollständig) | 07 |
| Information nach Anweisungsende               | 08 |
|                                               |    |
| Name mehrfach definiert                       | 10 |
| Variablenname fehlt                           | 11 |
| Variablenname ist unzulässig                  | 12 |
| Feldvereinbarung ist unzulässig               | 13 |
| Zeichen ist unzulässig; Komma fehlt           | 14 |
| Teilstrings sind nicht erlaubt                | 15 |
| Marke fehlt oder ist unzulässig               | 16 |
| Laufparameter muß einfache Variable sein      | 17 |
|                                               |    |
| Element noch nicht implementiert              | 20 |
| Fehler in Klammerstruktur                     | 21 |
| Operator hier unzulässig                      | 22 |
| Aufbau des Ausdrucks unzulässig               | 23 |
| Index- oder Parameteranzahl ist falsch        | 23 |
| Zahlenkonstante falsch oder unzulässig        | 25 |
| Keine Zuweisung im LET                        | 26 |
| Reserviertes Wort hier unzulässig             | 27 |
| Komma hier unzulässig                         | 28 |
|                                               |    |
| Abbruch wegen Speichermangel                  | A1 |
| Abbruch wegen Überlauf der Symboltabelle      | A2 |
| Abbruch wegen Überlauf der Sprungzieltabelle  | A3 |
| Abbruch wegen fehlerhafter Quelleingabe       | A4 |
| ISA innerhalb VSA                             | A5 |

Beispiel

40 PRINT A\$; B

14 ← ————— Fehlerhinweis  
 ? , ← ————— Fehler-Nr.



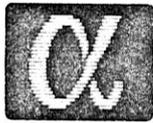
alphaTronic

# BASIC COMPILER

## Fehlermeldungen und Fehlercodes

### Fehlercodes der Generierungsphase

| BEDEUTUNG                                                                                              | CODE |
|--------------------------------------------------------------------------------------------------------|------|
| Leistung noch nicht implementiert                                                                      | C0   |
| Marke in der Sprungzieltabelle mehrfach vorhanden                                                      | C1   |
| Unbekanntes Sprungziel                                                                                 | C2   |
| Zu viele Feldelemente in einer DIM-Anweisung                                                           | C3   |
| Mixed-mode Fehler                                                                                      | C4   |
| oder                                                                                                   |      |
| Länge einer Stringvariablen für numerische Zuweisung zu klein                                          |      |
| Speichermangel während der Generierungsphase                                                           | C5   |
| Unzulässige Schachtelung von FOR-Schleifen                                                             | C6   |
| oder                                                                                                   |      |
| FOR-Anweisung ohne NEXT                                                                                |      |
| ONINT-Statement mehrfach mit gleicher Interrupt-Nummer vorhanden oder Interrupt-Nummer keine Konstante | C7   |
| Unerlaubte Stellung von Prozess- I/O-Variablen                                                         | C8   |
| Operator hier unzulässig                                                                               | C9   |



alphaTronic

# STICHWORTVERZEICHNIS

|   |                                  |         |
|---|----------------------------------|---------|
| A | ABS .....                        | 62      |
|   | AND .....                        | 62      |
|   | Anschlüsse .....                 | 14      |
|   | Arithmetische Operatoren .....   | 61      |
|   | ASSIGN .....                     | 31      |
|   | ATN .....                        | 62      |
| B | BASIC COMPILER .....             | 42-75   |
|   | — Arbeitsweise .....             | 44      |
|   | — Kommandos .....                | 24      |
|   | — Kommando-Syntax .....          | 46      |
|   | — laden .....                    | 43      |
|   | — MOS-Kommandos .....            | 46      |
|   | — Programmbefehle - Syntax ..... | 52      |
|   | — Speicherbelegung .....         | 44      |
|   | — Tastenfunktionen .....         | 45      |
|   | — Zeichenvorrat .....            | 45      |
|   | BATCH-MODE .....                 | 40      |
|   | Baugruppenordnung .....          | 10-13   |
|   | BEGIN .....                      | 49      |
|   | BIFX .....                       | 71      |
|   | Bildschirmanschluß .....         | 15      |
|   | BUS .....                        | 3,76-77 |
| C | CALCULATE .....                  | 51      |
|   | CALL-Routinen .....              | 66-72   |
|   | CHAR .....                       | 58      |
|   | CHG .....                        | 62      |
|   | CIS .....                        | 65      |
|   | CLOSE .....                      | 67      |
|   | COMMAND OUTPUT .....             | 38      |
|   | COMPARE MEMORY .....             | 39      |
|   | COMPILE .....                    | 49      |
|   | CON .....                        | 62      |
|   | COS .....                        | 62      |
|   | CPU .....                        | 78-80   |
|   | CRT-Interface .....              | 87-93   |
|   | CSB .....                        | 71      |
| D | DAS .....                        | 71      |
|   | DELETE .....                     | 48      |
|   | DEV .....                        | 64      |
|   | DIM .....                        | 58      |
|   | Diskette einlegen .....          | 18      |
|   | Disketten-Handling .....         | 104     |
|   | Disketten-Organisation .....     | 103     |
|   | DISPLAY MEMORY .....             | 38      |
|   | Drucker .....                    | 105     |
|   | Drucker-Anschluß .....           | 16      |
|   | DUMP .....                       | 48      |
|   | Duplizieren .....                | 20      |



alphaTronic

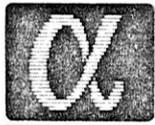
# STICHWORTVERZEICHNIS

|   |                               |           |
|---|-------------------------------|-----------|
| E | Eingabe-Fehler .....          | Anh.B2    |
|   |                               | 74        |
|   | END .....                     | 60        |
|   | ENDFILE .....                 | 36        |
|   | Eröffnen Systemgeräte .....   | 33-34     |
|   | EXP .....                     | 62        |
|   | Expansionsmöglichkeiten ..... | 4         |
|   | EXTERNAL .....                | 72        |
|   | EXTEND MOS .....              | 39        |
|   | E/A-Struktur .....            | 32        |
| F | Fehlercodes .....             | Anh.B1-B2 |
|   |                               | 73-75     |
|   | Fernsehgerät .....            | 15        |
|   | Festkommazahlen .....         | 55        |
|   | FIC .....                     | 70        |
|   | FILL MEMORY .....             | 39        |
|   | FINP .....                    | 67        |
|   | Floppy-Disk-Interface .....   | 99-100    |
|   | FMT .....                     | 68        |
|   | FOR/NEXT .....                | 59        |
|   | Formatieren .....             | 19        |
|   | Funktionen .....              | 62        |
| G | Generierungsfehler .....      | Anh. B3   |
|   |                               | 75        |
|   | Gerätezuordnung .....         | 31        |
|   | Gleitkommazahlen .....        | 54        |
|   | GO .....                      | 37        |
|   | GOSUB .....                   | 59        |
|   | GOTO .....                    | 59        |
| H | Hardware .....                | 76-107    |
|   | Hardwaregrundlagen .....      | 25        |
|   | HEX .....                     | 50        |
|   | Hexadezimalsystem .....       | 22        |
| I | IF .... THEN .....            | 60        |
|   | IFX .....                     | 62        |
|   | IO - Byte .....               | 41        |
|   | IOS .....                     | 32        |
|   | Inbetriebnahme .....          | 17-20     |
|   | INCW .....                    | 69        |
|   | Industriemonitor .....        | 15        |
|   | INF .....                     | 51        |
|   | Initialisieren .....          | 18        |
|   | INPUT .....                   | 36,65     |
|   | INT .....                     | 62        |
|   | INTEGER .....                 | 58        |
|   | Interface .....               | 8         |



# STICHWORTVERZEICHNIS

|    |                                     |         |
|----|-------------------------------------|---------|
| K  | KILO BYTE .....                     | 22      |
|    | Kommandos .....                     | 24      |
|    | Konfigurationsmöglichkeiten .....   | 8-13    |
| L  | LET .....                           | 58      |
|    | LGA .....                           | 62      |
|    | Library-Fehler .....                | Anh.B1  |
|    |                                     | 73      |
|    | LINK .....                          | 51      |
|    | LIST .....                          | 48      |
|    | LIST COMMAND OUTPUT .....           | 38      |
|    | Literaturhinweise .....             | 7       |
|    | LOAD .....                          | 47      |
|    | LOG .....                           | 62      |
|    | Logische Operatoren .....           | 61      |
| M. | Mathematische Funktionen .....      | 62      |
|    | MAX .....                           | 62      |
|    | MIN .....                           | 62      |
|    | Mini-Floppy-Drive .....             | 101-102 |
|    | Mini-Floppy-Interface .....         | 99-100  |
|    | MOS-Kommandos .....                 | 24,30   |
|    | MOS-Monitor .....                   | 27-41   |
|    | MOVE .....                          | 70      |
|    | MOVE MEMORY .....                   | 38      |
| N  | NETZTEIL .....                      | 97-98   |
|    | NEW .....                           | 47      |
|    | NEXT .....                          | 59      |
|    | NOT .....                           | 62      |
| O  | OLD .....                           | 47      |
|    | OPEN .....                          | 66      |
|    | Operatoren .....                    | 61      |
|    | OR .....                            | 62      |
|    | OUTPUT .....                        | 35      |
| P  | Peripherie .....                    | 101-107 |
|    | Pin-Belegung BUS .....              | 76      |
|    | Pin-Belegung Tastatur IF .....      | 84      |
|    | Pin-Belegung CRT-IF .....           | 88      |
|    | Pin-Belegung 16 K RAM-Platine ..... | 95      |
|    | Pin Belegung Mini-Floppy IF .....   | 100     |
|    | Port's .....                        | 69      |
|    | PUT CW .....                        | 69      |
|    | PRESET DEVICE .....                 | 33-34   |
|    | PRINT .....                         | 63-65   |
|    | PRINT DEV .....                     | 64      |
|    | PRINT TAB .....                     | 65      |
|    | Programmbefehle .....               | 24      |
|    | Programmbibliothek .....            | 6       |
|    | Programme laden .....               | 22-23   |



alphaTronic

# STICHWORTVERZEICHNIS

|   |                               |         |
|---|-------------------------------|---------|
| Q | Q .....                       | 51      |
| R | RAM .....                     | 23      |
|   | RAM Platine 16 K .....        | 94-96   |
|   | Rangfolge .....               | 62      |
|   | REM .....                     | 60      |
|   | RENUMBER .....                | 48      |
|   | RESET .....                   | 17      |
|   | RETURN .....                  | 59,72   |
|   | ROM .....                     | 23      |
|   | RUN .....                     | 50      |
|   | Runden .....                  | 71      |
| S | SAVE .....                    | 47      |
|   | Spezielle Schnittstelle ..... | 80      |
|   | SIN .....                     | 62      |
|   | Software .....                | 5,27-75 |
|   | Softwaregrundlagen .....      | 26      |
|   | Sonderfunktionen .....        | 62      |
|   | Speicher .....                | 22      |
|   | Speicherbelegung .....        | 29      |
|   | Speichermedien .....          | 23      |
|   | Sprachumfang (BASIC) .....    | 53      |
|   | SQR .....                     | 62      |
|   | START .....                   | 50      |
|   | String's .....                | 55      |
|   | Stringvariablen .....         | 57      |
|   | Stromversorgung .....         | 97-98   |
|   | SUBROUTINE .....              | 72      |
|   | SUBSTITUTE .....              | 37      |
|   | Syntax, Basic-Kommandos ..... | 46      |
|   | Syntax, MOS-Kommandos .....   | 30      |
|   | SYSIO .....                   | 69      |
|   | Systemdienste .....           | 51      |
| T | TAB .....                     | 65      |
|   | TAN .....                     | 62      |
|   | Tastatur .....                | 81      |
|   | Tastatur-Interface .....      | 83-86   |
|   | Tastencode .....              | 82      |
|   | Tastenfunktionen .....        | 21,45   |
|   | Texthaltung .....             | 47-48   |
|   | THEN .....                    | 60      |
|   | TOP OF MEMORY .....           | 39      |
| U | USER PROGRAM .....            | 40      |
| V | Variablen .....               | 54      |
|   | Vergleichs-Operatoren .....   | 61      |
| W | Warnungen .....               | Anh.B1  |
|   |                               | 73      |



alphaTronic

## STICHWORTVERZEICHNIS

|   |                      |    |
|---|----------------------|----|
| X | X .....              | 51 |
|   | XDR .....            | 62 |
| Z | Zahlen .....         | 54 |
|   | Zahlenvariable ..... | 56 |
|   | Zeichen .....        | 54 |
|   | Zeichenvorrat .....  | 45 |