

BILDUNGSCOMPUTER
robotron A5105

SCP-Handbuch
Teil 2

VEB ROBOTRON-MESSELEKTRONIK >OTTO SCHÖN< DRESDEN
Lingnerallee 3, Postfach 211, Dresden, DDR-8012

I N H A L T

Teil 2

8.	Grundlagen der Programmierung	62
9.	Beschreibung der BDOS-Funktionen	65
10.	Ein Beispiel für die Anwendung der BDOS-Funktionen	84
11.	Beschreibung der BIOS-Funktionen	87
ANHANG C	Reservierte Speicherplätze	98
ANHANG D	BDOS-Funktionen	99
ANHANG E	BIOS-Funktionen	102
ANHANG F	Allgemeine Fehlermeldungen	103
	Sachwortverzeichnis	106

5/89 c

digitalisiert: Ulrich Zander 05/2002 <zander@felix.sax.de>

VEB ROBOTRON-MESSELEKTRONIK >OTTO SCHÖN< DRESDEN
Lingnerallee 3, Postfach 211, Dresden, DDR-8012

Ausführlicher Inhalt (Teil 2)

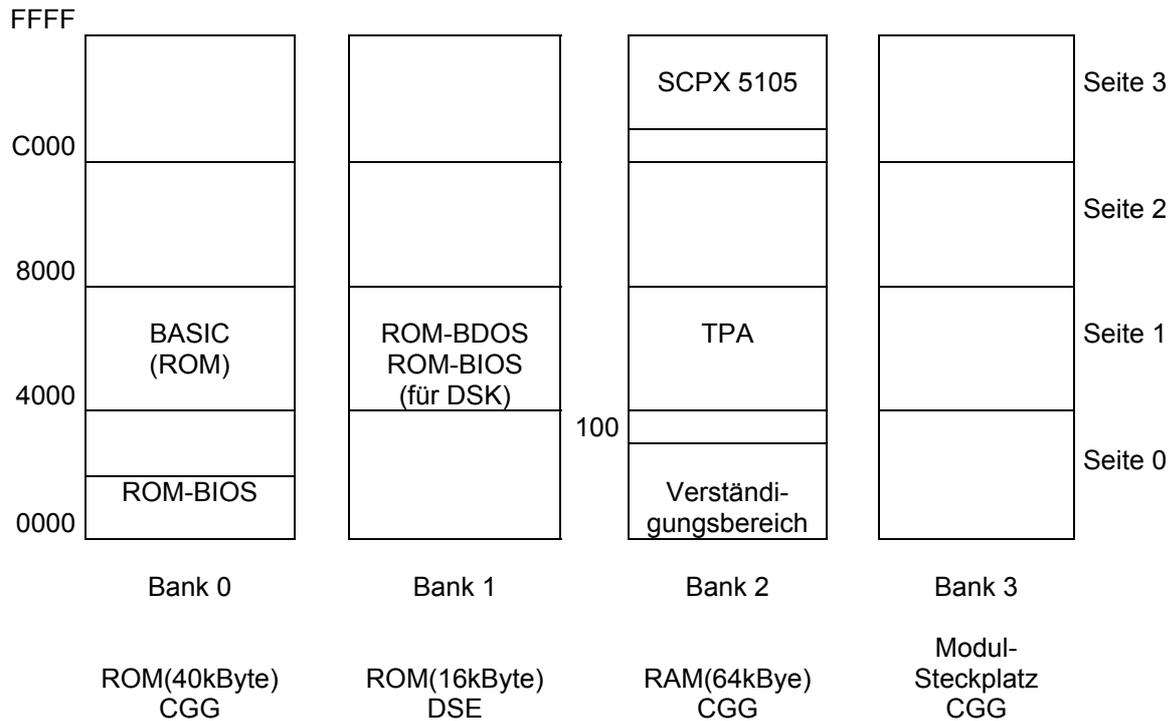
8.	Grundlagen der Programmierung	62
8.1.	Speicherkonzept	62
8.2.	Logische Geräte	64
8.3.	Kommandoeingabe über CCP	64
9.	Beschreibung der BDOS-Funktionen	65
9.1.	Aufgaben des BDOS	65
9.2.	Schnittstelle des BDOS	66
9.3.	Zeichenorientierte Ein- und Ausgabe	67
9.3.1.	Zuordnung der Kanäle	67
9.3.2.	Datenverkehr über die Konsole (CON)	67
9.3.3.	Zeichenweiser Datenverkehr über die logischen Geräte (RDR, PUN, LST)	70
9.4.	Arbeit mit Diskettendateien	71
9.4.1.	Aufbau des FCB	71
9.4.2.	Verwaltung des Inhaltsverzeichnisses	72
9.4.3.	Adressierung und Datentransport	75
9.4.4.	System- und Hilfsfunktionen	78
10.	Ein Beispiel für die Anwendung der BDOS-Funktionen	84
11.	Beschreibung der BIOS-Funktionen	87
11.1.	Aufgaben des BIOS	87
11.2.	Schnittstellen des BIOS	87
11.3.	Initialisieren des SCPX 5105 (BOOT, WBOOT)	88
11.4.	Zeichenweiser Datenverkehr über die logischen Geräte	89
11.4.1.	Logische Ein-/Ausgabekanäle	89
11.4.2.	BIOS-Schnittstellen für diese Geräte	91
11.5.	Spezielle Arbeit mit der Diskette	92
11.5.1.	Laufwerkssteuerung (HOME, SELDSK, SETTRK, SETSEC, SECTRAN)	92
11.5.2.	Datenverkehr (SETDMA, READ, WRITE)	94
11.6.	Beschreibung der Diskettenformate	95
11.6.1.	Diskettenparameterkopf DPH	96
11.6.2.	Diskettenparameterblock DPB	96
ANHANG C	Reservierte Speicherplätze	98
ANHANG D	BDOS-Funktionen	99
	- BDOS-Funktionen	99
	- Aufbau des FCB	101
	- Aufbau der LW-Vektoren	101
ANHANG E	BIOS-Funktionen	102
ANHANG F	V.24-Treiber im SCPX 5105	103
	- Druckertreiber	103
	- Allgemeiner V.24-Treiber	104
	- Fehlermeldungen	104
	Sachwortverzeichnis	106

8. Grundlagen der Programmierung

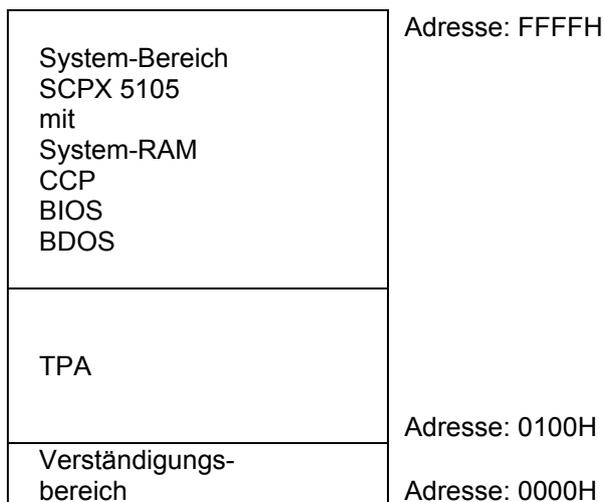
8.1. Speicherkonzept

Damit unter SCPX 5105 ein möglichst großer Arbeitsspeicher (TPA, s.u.) zur Verfügung steht, wurden viele Teile des ROM-Betriebssystems in das SCPX 5105 einbezogen. Das trifft besonders auf das ROM-BIOS mit den Treibern für Diskette, Bildschirm, Tastatur sowie Drucker und auf das ROM-BDOS für die logische Verwaltung der Diskette zu.

Daraus ergibt sich folgende Nutzung der im BILDUNGSCOMPUTER robotron A 5105 zur Verfügung stehenden vier 64-kByte-Speicherbänke (slots):



Nach dem Initialisieren des SCPX 5105 ("Kaltstart") ist der 64-kByte-Speicher, der für Nutzerprogramme verfügbar ist (Bank 2), in folgende Bereiche aufgeteilt:



In der vorangegangenen Darstellung bedeuten:

BIOS Basic-I/O-System

Dieser Systembereich enthält die Verbindung zu den hardwarespezifischen Programmteilen des ROM-BIOS. Am Anfang steht eine Sprungtabelle, die zu den einzelnen Routinen zur Bedienung der im System eingebundenen physischen E/A-Geräte führt (vgl. Abschnitt 11).

BDOS Basic Disc Operating System

Dieser Teil des Systems bietet dem Programmierer alle Routinen an, die den problemlosen Datenaustausch mit den logischen Geräten gewährleisten. Zur Nutzung dieser Routinen, auch BDOS-Funktionen genannt, gibt es ein standardisiertes Verfahren zu deren Aufruf. Das BDOS des SCPX 5105 bedient sich der physischen BIOS-Routinen und der Routinen des ROM-BDOS (vgl. Abschnitte 9 und 10).

CCP Consol Command Processor

Nach "Kaltstart" bzw. Systemneustart übernimmt CCP die Steuerung des Betriebssystems. In einem Systemteil sind die residenten Kommandos DIR, ERA, TYPE, SAVE, REN, USER, EXT, RES, SWAP, HELP und GO realisiert, die den TPA nicht verändern (vgl. Abschnitt 4.1).

Außerdem können mit CCP auch transiente Kommandos aufgerufen werden. Diese Programme werden in den TPA geladen. Dann übergibt CCP die Steuerung an die transienten Programme. Nach Abarbeitung wird die Steuerung an das CCP zurückgegeben (vgl. Abschnitte 4.2 und 8.3).

TPA Transient Program Area

Dies ist der Speicherbereich, der den transienten Kommandos und Programmen zur Verfügung steht und bei der Adresse 0100H beginnt. Ab dieser Adresse werden die transienten Kommandos oder Programme geladen, die dann auch bei 0100H gestartet werden. Dieser Bereich steht ebenfalls für den Anwenderprogrammierer zur Verfügung. Die Größe des TPA beträgt ca. 57 kBytes.

Verständigungsbereich:

In diesem Bereich sind Adressen, Kennbytes, Sprünge und Standardpuffer für die Arbeit des SCPX 5105 abgelegt (vgl. Anhang C).

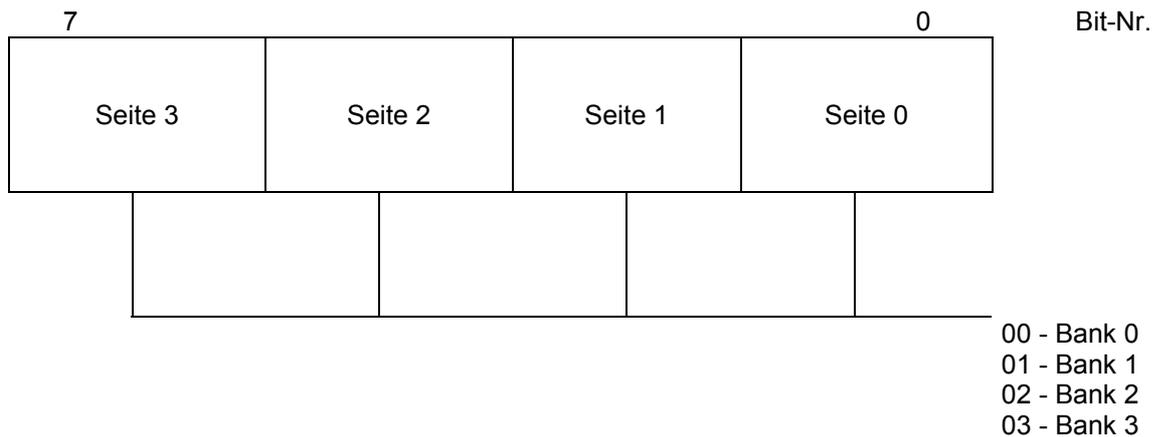
Anwenderprogramme, die ausschließlich im TPA laufen und nur auf die Systemschnittstellen des SCPX 5105 zugreifen - d.h. nur BDOS- und BIOS-Rufe verwenden - , bleiben von notwendigen Bankumschaltungen, die das SCPX 5105 selbst vornimmt, unbeeinflusst.

Sollen aus Anwenderprogrammen ROM-BIOS-Routinen direkt aufgerufen werden (z.B. Routinen zur grafischen Ausgabe auf den Bildschirm, Seite 0, Bank 0), so muß die notwendige Speicherkonfiguration von diesen Anwenderprogrammen selbst eingestellt werden.

Dabei ist zu beachten:

- Es ist immer eine aktuelle Speicherkonfiguration von 64 kBytes eingeschaltet. Sie besteht aus jeweils einer der Seiten 0 bis 3 aus den Bänken 0 bis 3. Für die Anwenderprogramme des SCPX 5105 ist das immer die vollständige Bank 2.
- Das "Wurzelsegment bildet Seite 3, Bank 2. Es enthält die Eintrittspunkte für BDOS und BIOS, den CCP und den System-RAM-Bereich mit dem Interruptvektor. Es darf nicht weggeschaltet werden.
- Die gerade aktive Seite (entsprechend PC-Stand) darf nicht weggeschaltet werden. Sollen Routinen auf gleichen Seiten anderer Bänke gerufen werden, so muß das über einen "Umweg" über eine andere Seite, z.B. Wurzelsegment Seite 3, Bank 2, erfolgen.

Die Einstellung oder Abfrage der Speicherkonfiguration erfolgt über Port A des Speicherverwaltungsschaltkreises SVG, Adresse 0A8H (vgl. Anhang D zum Programmierhandbuch RBASIC):



Für die Anwenderprogramme des SCPX 5105 gilt also immer
Port 0A8H, SVG = 0AAH, Seite 0 bis 3 auf Bank 2.

Spezielle Hinweise zur Einbindung von zusätzlichen Interruptroutinen und nachladbaren Treibern, die im Interruptbetrieb arbeiten sollen, entnehmen Sie bitte den entsprechenden Beschreibungen.

8.2. Logische Geräte

Vom SCPX 5105 werden folgende logische Geräte unterstützt:

- die Konsole für Dialogverkehr (standardmäßig Tastatur und Bildschirm)
- ein Listgerät (standardmäßig V.24-Drucker am Anschluß PRINTER)
- Eingabe und Ausgabe über sequentielle Datenkanäle (standardmäßig: V.24-Eingabe/Ausgabe am Anschluß PLOTTER).
aktiviert; z.B. V.24-Anschluß für PLOTTER durch Nachladen des Treibers möglich.

Um mit den logischen Geräten arbeiten zu können, sind die BDOS bzw. BIOS-Rufe zu verwenden (siehe Abschnitte 9 und 11).

Die Zuweisung anderer Treiberprogramme als der standardmäßig eingestellten, erfolgt mit dem transienten Kommando STAT (siehe Abschnitt 4.2.3). Andere Treiberprogramme, die nicht Bestandteil des ROM-Betriebssystems SCPX 5105 sind, können mittels Nachladeprogramm installiert werden.

8.3. Kommandoeingabe über CCP

Ein Kommando-/Programmaufruf erfolgt stets durch die Eingabe des residenten Kommandos oder des Dateinamens des transienten Programmes. Der Dateityp COM des transienten Programms braucht nicht eingegeben zu werden. Danach ist noch die Eingabe von Parametern möglich. Diese werden generell auf den Adressen 0081H bis 00FH abgelegt. Es sind somit insgesamt 127 Zeichen bei der Parametereingabe möglich; die tatsächliche Anzahl eingegebener Zeichen wird auf der Adresse 0080H abgelegt.

Der Programmierer muß aber darauf achten, daß die Adresse 0080H nach einem Kalt- und Warmstart die aktuelle DMA-Adresse für Floppy-Disk-Zugriffe ist. Werden die Parameter noch nach einem FD-Zugriff benötigt, muß der Programmierer diese Parameter in einen anderen Speicherbereich verschieben oder eine andere DMA-Adresse definieren (siehe Abschnitt 9.4, Funktion 26).

Neben der Speicherung der Parameter ab 0080H werden außerdem die direkt dem Kommando bzw. Programmnamen möglicherweise folgenden zwei Dateibezeichnungen (Laufwerkscode, Dateiname, Dateityp) auf 005CH und 006CH abgelegt. Sind Name und Typ der Datei kürzer als 11 Zeichen, werden die freien Bytes mit Leerzeichen (20H) belegt. Zwischen Dateiname und -typ ist bei der Eingabe ein Punkt einzugeben.

Die Adresse 005CH ist aber auch als Standardadresse für den FCB nach Kalt- und Warmstart definiert. Dies hat der Programmierer in gleicher Weise zu beachten wie die Nutzung des Speicherbereichs ab 0080H.

Bei der Speicherung der Parameter ab 005CH, 006CH und 0080H werden generell alle eingegebenen Buchstaben als Großbuchstaben abgelegt.

Beispiel: A>TEST 123.abc B:5.DE fghENTER

Es wird gespeichert (alle Zahlenangaben hexadezimal):

005C:	00		Laufwerkscode
005D:	31 32 33 20 20 20 20 20	123	Dateiname
0065:	41 42 43	ABC	Dateityp
006C:	02	.	Laufwerkscode
006D:	35 20 20 20 20 20 20 20	5	Dateiname
0075:	44 45 20	DE	Dateityp
0080:	13		Anzahl gültiger Zeichen
0081:	20 31 32 33 2E 41 42 43	123.ABC	eingegebenes Kommando
0089:	20 42 3A 35 2E 44 45 20	B:5.DE	
0091:	46 47 48	FGH	

Ein Kommando- oder Programmaufruf erfolgt vom CCP-Status aus mit CALL, d.h. das Kommando bzw. Programm kann mit RET zu CCP zurückkehren. Der Programmierer muß aber unbedingt beachten:

- Der von CCP bereitgestellte Stack ist beim SCPX 5105 genau 30H Bytes groß, wobei bereits die adreßhöchsten Bytes die Rückkehradresse zum CCP enthalten.
- Dieser Stack reicht für die meisten Anwendungsfälle aus. Auch bei BDOS- und BIOS-Rufen besteht keine Gefahr für Stacküberläufe, weil diese Aufrufe beim SCPX 5105 einen eigenen Stack einrichten.
An dieser Stelle muß ausdrücklich darauf hingewiesen werden, daß diese Bemerkungen nur für SCPX 5105 gelten, andere SCP-Betriebssysteme stellen einen wesentlich kleineren Stack bereit und belasten diesen auch bei BIOS-Rufen.
- In jedem Programm kann auch ein eigener Stack definiert werden. Vor dem RET zur Rückkehr in das CCP ist dann der alte Stackpointer wieder einzustellen. Andernfalls ist das Programm unbedingt mit einem Warmstart zu beenden, damit CCP neu aktiviert wird. Dadurch wird der CCP-Status wieder erreicht, ohne daß der alte Stackpointerzustand eingestellt werden muß.
- Es ist zweckmäßig, das Anlegen eines programm eigenen Stacks und das Beenden eines Programms mit dem Warmstart generell vorzusehen.
- Ein Nachladeprogramm für Treiber muß grundsätzlich mit RET enden, sonst wird der Treiber vom System nicht erkannt, und sein Speicherbereich nicht vor dem Überschreiben geschützt.

9. Beschreibung der BDOS-Funktionen

9.1. Aufgaben des BDOS

Das Diskettengrundbetriebssystem (BDOS) stellt den Kern des SCPX-Betriebssystems dar. Es unterstützt die Ein- und Ausgabe auf den logischen Geräten (Konsole, Listgerät, sequentielle Datenkanäle) sowie die Dateiarbeit auf den Disketten.

Über BDOS-Ruf können folgende Aufträge von BDOS angefordert werden:

- Zeichenorientierte Ein- und Ausgabe
 - Zuordnung der logischen zu den physischen Ein- und Ausgabekanälen
 - zeichenweise Ein- und Ausgabe über die Konsole

- zeichenweise Ein- und Ausgabe über die sequentiellen Datenkanäle
- zeichenweise Ausgabe über den Listkanal

- Arbeit mit Diskettendateien
 - allgemeine Dateimanipulationen und -verwaltung
 - sequentieller Dateizugriff
 - direkter Dateizugriff

- Systemverwaltung
 - Initialisierung
 - Ermittlung des Systemzustandes
 - weitere Hilfsfunktionen

9.2. Schnittstelle des BDOS

Der BDOS-Ruf erfolgt über einen Unterprogrammaufruf auf der absoluten Speicheradresse 5. Dieser Speicherbereich liegt im Verständigungsbereich. An dieser Stelle befindet sich ein Sprungbefehl, der dort bei der Systeminitialisierung eingetragen wird. Dieser Sprungbefehl setzt im BDOS-Körper auf. Der Ruf muß noch durch Belegung von Registern spezifiziert werden.

Die BDOS-Funktionen sind mit laufenden Nummern versehen. Im CPU-Register C muß vor Ausführung die Codenummer eingestellt werden, die der gewünschten BDOS-Funktion zugeordnet ist.

Die zusätzlichen Eingabeparameter werden bei 8-Bit-Werten in das CPU-Register E, bei 16-Bit-Werten in das CPU-Doppelregister DE eingetragen. Nach Ausführung der BDOS-Funktion wird das aufrufende Programm hinter der Aufrufstelle fortgesetzt. Rückgabeparameter nach Abarbeitung der BDOS-Funktion werden als

- 8-Bit-Werte im CPU-Register A,
- 16-Bit-Werte im CPU-Doppelregister HL,
- Feldeintragen in einem Feld, das durch das CPU-Doppelregister DE vor Aufruf adressiert wurde,

bereitgestellt.

Das Betriebssystem benutzt einen eigenen Stapel, so daß das aufrufende Programm nur mit einer Stapel-Ebene durch den Aufruf belastet wird.

9.3. Zeichenorientierte Ein- und Ausgabe

9.3.1. Zuordnung der Kanäle

Der zeichenorientierte Datenaustausch mit den Peripheriegeräten erfolgt über die 4 logischen Kanäle:

- Konsole
- sequentielle Eingabe
- sequentielle Ausgabe
- Listgerät.

Jedem dieser Kanäle kann einer von jeweils 4 kanalspezifischen physischen Subkanälen zugeordnet werden. Dieses Zuordnungsbyte (I/O-Byte) befindet sich im Verständigungsbereich. Die Zuordnung der Subkanäle geschieht in folgender Weise:

- BIT 0 und 1 : Konsole
- BIT 2 und 3 : sequentielle Eingabe
- BIT 4 und 5 : sequentielle Ausgabe
- BIT 6 und 7 : Listgerät

Funktion 7*): Abfragen des I/O-Bytes

Aufrufparameter:
Register C: 07H

Rückgabewert:
Register A: I/O-Byte-Wert

Bereitstellung des aktuellen Wertes des I/O-Bytes im CPU-Register A.

Funktion 8: Setzen des I/O-Bytes

Aufrufparameter:
Register C: 08H
Register E: I/O-Byte-Wert

Die Belegung des CPU-Registers E wird in das I/O-Byte eingetragen.

9.3.2. Datenverkehr über die Konsole (CON)

Der Dialogverkehr zwischen Gerät und Bediener erfolgt über die Konsole.

Es sind die 2 Betriebsarten

- zeichenweise Ein-/Ausgabe
- gepufferte Ein-/Ausgabe

möglich.

*) Die Funktionsnummern beziehen sich auf die Nummern der BDOS-Funktionen (vgl. auch Anhang D).

Der zeichenweise Datenverkehr kann entweder im BDOS-gesteuerten oder im BIOS-gesteuerten Modus erfolgen. Im ersten Modus erfolgt bei der Eingabe der Zeichen eine Echofunktion auf der Konsole sowie eine Sonderbehandlung folgender Eingabebelegungen:

- ^I:** Tabulation auf den nächsten 8-Spalten-Tabulator
- STOP:** Die nächste Konsoleneingabe wird suspendiert, bis ein beliebiges Zeichen von der Konsole eingegeben wird.
Der BDOS-Ruf wird im Wartezustand nicht verlassen, d.h. das rufende Programm nicht fortgesetzt.
- ^P:** Die nachfolgenden Ausgaben von Konsolezeichen erfolgen über die Kanäle Konsole und Listgerät parallel. Durch Eingabe von ^P wird diese Funktion wieder zurückgesetzt (triggende Funktion).

<p>Funktion 1: Konsolen-Eingabe</p> <p>Aufrufparameter: Register C: 01H</p> <p>Rückgabewert: Register A: Zeichen</p>

Das nächste Zeichen von der Konsole wird in das Register A übergeben. In der aufgerufenen Routine wird solange verblieben, bis das nächste Zeichen von der Konsole zur Verfügung gestellt werden kann. Die eingegebenen Zeichen werden als Echo auf der Konsole ausgegeben. STOP und die Steuerzeichen ^P und ^I werden ausgewertet; ^C wird nicht ausgefiltert.

<p>Funktion 2: Konsolen-Ausgabe</p> <p>Aufrufparameter: Register C: 02H</p> <p>Rückgabewert: Register E: Zeichen</p>

Die Belegung des CPU-Registers E wird auf der Konsole ausgegeben. Das Steuerzeichen ^P als Ausgabezeichen wird ignoriert, ^I wird berücksichtigt. STOP und das über die Konsole eingegebene Zeichen ^P werden bei der Ausgabe interpretiert.

<p>Funktion 2: Konsolen-Status</p> <p>Aufrufparameter: Register C: 0BH</p> <p>Rückgabewert: Register A: Status</p>

Die Funktion prüft, ob bereits das nächste Zeichen auf der Konsole eingegeben worden ist. Die Belegung des CPU-Registers A bei Rückkehr ist mit

- A = 00H keine Zeicheneingabe
- A = 01H Zeicheneingabe erfolgt

zu interpretieren. Das eingegebene Zeichen verursacht solange den Belegt-Status, bis das Zeichen über Konsoleneingabe abgerufen wird.

Die Steuerzeichen ^S und ^C werden als eingegebene Zeichen gewertet.

Funktion 6: Konsolenein- und ausgabe
direkt

Aufrufparameter:

Register C: 06H

Register E: 0FFH (Eingabe) oder Zeichen (Ausgabe)

Rückgabewert:

Register A: Zeichen oder Status

Enthält das CPU-Register E den Wert 0FFH, so erfolgt eine Konsoleneingabe in das CPU-Register A. Wurde kein Zeichen über die Konsole eingegeben, wird das CPU-Register A mit dem Code 00H belegt. Es erfolgt keine Pufferung der Eingabedaten, sondern eine kombinierte Eingabe-/Statusfunktion wird realisiert. Enthält das CPU-Register E einen von 0FFH verschiedenen Wert, wird dieses Zeichen auf der Konsole ausgegeben.

Funktion 9: Ausgabe einer Zeichenkette auf Konsole

Aufrufparameter:

Register C: 09H

Register DE: Adresse der Zeichenkette

Der Inhalt des CPU-Registers DE wird als Adresse auf der Zeichenkette interpretiert. Die Zeichenkette kann beliebige Zeichen enthalten. Das erste auftretende Zeichen \$ wird als Endekennzeichen der Zeichenkette interpretiert und nicht mit ausgegeben. BDOS-Zustände, die über ^P oder ^I gesteuert werden, werden berücksichtigt.

Funktion 10: Eingabe in den Konsolenpuffer

Aufrufparameter:

Register C: 0AH

Register DE: Pufferadresse

Rückgabewert:

Zeichen von Konsole im Puffer

Diese Funktion dient der Eingabe einer ganzen Zeile in einen durch das Registerpaar DE adressierten Puffer, wobei gleichzeitig die Anzeige der eingegebenen Zeichen auf der Konsole erfolgt.

Das 1. Byte des Puffers muß die maximal eingebbare Zeichenzahl enthalten (1 bis 255).

Das 2. Byte des Puffers enthält nach Rückkehr die tatsächliche Anzahl eingegebener Zeichen und definiert den Gültigkeitsbereich des Puffers.

Ab der 3. Position des Puffers stehen die eingegebenen Zeichen. Der Puffer wird vor der Eingabe gelöscht, so daß nach dem zuletzt eingegebenen Zeichen beliebige Zeichen folgen.

Die Zeichenausgabe wird beendet durch:

- Erreichen der Maximalzahl (1. Byte des Puffers),
- Eingabe der Abschlußtaste ENTER,

- Eingabe ^J (siehe Anhang B),
- Eingabe ^M (siehe Anhang B).

Dabei werden ^J und ^M nicht in den Eingabepuffer übernommen.

Der Cursor der Konsole wird danach auf den Anfang der laufenden Konsolenzeile positioniert.

Eine Reihe von Steuerzeichen ermöglichen ein einfaches zeilenorientiertes Editieren:

DEL, ^H, ^S: Das letzte Zeichen im Puffer und auf der Konsole wird gelöscht.

^C: Wird dieses Zeichen am Pufferanfang eingeschrieben, so wird ein Warmstart (vgl. auch Abschnitt 11,3) ausgelöst. An anderen Positionen bewirkt dieses Steuerzeichen die Ablage des Codes (03H) im Puffer und die Anzeige ^C auf der Konsole (entspricht CTRL STOP).

^E: Dieses Steuerzeichen gelangt nicht in den Puffer, es bewirkt auf der Konsole eine Zeilenschaltung und eine Positionierung des Cursors an den Zeilenanfang (kein Eingabeende!).

^X: Der Puffer wird gelöscht, ebenfalls alle eingegebenen Zeichen auf der Konsole. Der Cursor nimmt die Position ein, die er vor dem Funktionsaufruf hatte.

^I: In den Puffer wird der Code (09H) eingetragen. Auf der Konsole werden so viele Leerzeichen ausgegeben, daß eine 8-Zeichen-Tabulation entsteht (Spalten 1,9,17,25...).

^P: Durch dieses Steuerzeichen wird bewirkt, daß jedes Zeichen nicht nur zur Konsole sondern auch zum Listgerät ausgegeben wird. Diese Parallelausgabe bleibt über die BDOS-Funktion #10 hinaus erhalten, so daß bei folgenden Funktionen #2, #9 und #10 eine Ausgabe auf Konsole und Listgerät erfolgt.

Die zusätzliche Ausgabe auf das Listgerät kann durch eine erneute Eingabe von ^P innerhalb der BDOS-Funktion #10 abgeschaltet werden.

9.3.3. Zeichenweiser Datenverkehr über die logischen Geräte (RDR, PUN, LST)

Funktion 3: sequentielle Eingabe	
Aufrufparameter:	
Register C:	03H
Rückgabewert:	
Register A:	Zeichen

BDOS wartet auf die Eingabe eines Zeichens über den sequentiellen Kanal und übergibt das Zeichen im CPU-Register A.

Funktion 4: sequentielle Ausgabe	
Aufrufparameter:	
Register C:	04H
Register E:	Zeichen

Ausgabe eines Zeichens aus dem CPU-Register E auf den sequentiellen Kanal.

Funktion 5: Ausgabe auf Listgerät

Aufrufparameter:

Register C: 05H

Register E: Zeichen

Das Zeichen aus dem CPU-Register E wird auf dem Listgerät ausgegeben.

9.4. Arbeit mit Diskettendateien

Bei fast allen diesen Funktionen wird ein Parameterblock benötigt, der alle Informationen zur Diskettenarbeit enthält. Dieser Block wird Dateisteuerblock (FCB) genannt. Meist ist seine Adresse vor dem BDOS-Aufruf im Registerpaar DE einzustellen.

Die gesamte Diskette ist logisch in Sätze mit je 128 Bytes aufgeteilt. Bei Datenmanipulation mit der Diskette kann immer nur auf diese Sätze zugegriffen werden. BDOS unterstützt die Organisation von Dateien, die aus einer Menge dieser Sätze bestehen. Alle Parameter der Dateien und für den Zugriff befinden sich im FCB.

9.4.1. Aufbau des FCB

Byte	Kurzbezeichnung	Bedeutung
+0	dr	Laufwerkscode 0 ... aktuelles Laufwerk (siehe auch Funktion 14) 1 ... LW A 2 ... LW B 3 ... LW C 16 ... LW P
+1...8	f1...f8	Dateiname Die siebenten Bits dieser Bytes sind für den Dateinamen nicht signifikant. Hier können Nutzerflags f...1 bis f...8 angelegt werden.
+9...11	t1...t3	Dateityp Die Bits 7 von t1 und t2 haben folgende Bedeutung: Bit 7 von t1 = 1 ---> die Datei ist Nur-Lese-Datei (R/O). Bit 7 von t2 = 1 ---> die Datei ist nur eine SYS-Datei (vgl. Abschnitt 3.6)
12	ex	aktuelle Bereichs-(Extent-)Nummer
13,14		für SCPX reserviert
15	rc	Füllstand des aktuellen Extent (Wertebereich: 0 bis 128) in Anzahl Sätzen
16...31		für SCPX reserviert
32	cr	aktuelle Satznummer des Extent für den nächsten sequentiellen Zugriff
33-35	r0,r1,r2	aktuelle Satznummer der Datei (nur für Direktzugriff erforderlich!) r0- L-Teil der Satznummer r1- H-Teil der Satznummer (Wertebereich: 0...65535) r2- für Überlauf

9.4.2. Verwaltung des Inhaltsverzeichnisses

Zur Verwaltung der Dateien befindet sich auf der Diskette ein Bereich mit einem Verzeichnis, in das die aktuellen Zeiger der Dateien eingetragen sind. Beim Zugriff auf Dateien werden, von diesen Eintragungen ausgehend, die Dateien manipuliert.

Zur Verwaltung dieser Eintragungen sind die folgenden BDOS-Funktionen installiert:

<p>Funktion 22: Anlegen Datei</p> <p>Aufrufparameter: Register C: 16H Register DE: FCB-Adresse</p> <p>Rückgabewert: Register A: Verzeichniscode</p>
--

Ist eine neue Datei anzulegen, so ist diese Funktion zu verwenden. Die BDOS-Funktion #15 (OPEN) ist nur für bereits existierende Dateien wirksam! Im Registerpaar DE ist die FCB-Adresse vorzugeben.

Da bei dieser Funktion keine Kontrolle auf eine bereits unter gleicher Bezeichnung bestehende Datei erfolgt, sollte sicherheitshalber vorher die BDOS-Funktion #19 (Löschen einer Datei) ausgeführt werden.

Andernfalls kann eine Doppeleintragung einer Datei mit gleichem Namen entstehen.

Die Datei wurde erfolgreich in den Verzeichnis angelegt, wenn im Register A als Rückmeldung der Wert 00H bis 03H steht. Eine 0FFH im CPU-Register A sagt aus, daß in dem Verzeichnis keine Eintragung mehr frei ist.

Nach der Funktion #22 ist die Funktion #15 (OPEN) nicht erforderlich!

<p>Funktion 15: Eröffnen einer Datei</p> <p>Aufrufparameter: Register C: 0FH Register DE: FCB-Adresse</p> <p>Rückgabewert: Register A: Verzeichniscode</p>

Im Registerpaar DE steht die Adresse des FCB. Mit dieser Funktion wird der im FCB benannte Bereich der Datei (Bytes f1...f8, t1...t3, ex), der auf dem durch das Byte 0 des FCB ausgewählten Laufwerk existieren muß, aktiviert. Dabei werden bestimmte Werte aus der Verzeichniseintragung in den FCB übernommen, so daß anschließend die Funktionen für das Lesen und Schreiben ausgeführt werden können.

Wird der Bereich der Datei gefunden, so wird im Register A der Wert 00H, 01H, 02H oder 03H gesetzt. Enthält das Register A den Wert 0FFH, wurde der Suchbegriff nicht gefunden.

Im Namen der Datei im FCB können auch Fragezeichen (/Codierung 3FH) enthalten sein. Für sie kann jedes andere beliebige Zeichen im Namen der Datei in dem Verzeichnis an der gleichen Position stehen. Es wird die erste Datei in dem Verzeichnis aktiviert, die mit den Zeichen in den anderen Positionen übereinstimmt.

Soll nach dem Eröffnen des Bereiches der Datei sequentiell auf den ersten Satz zugegriffen werden, so ist das Byte "cr" im FCB auf 00H zu setzen.

Der erste Bereich einer Datei muß immer Bereich Null (ex=0) sein.

Funktion 16: Schließen einer Datei

Aufrufparameter:

Register C: 10H
Register DE: FCB-Adresse

Rückgabewert:

Register A: Verzeichniscode

Im Registerpaar DE ist die Adresse des FCB anzugeben. Mit dieser Funktion werden die im FCB gespeicherten Informationen in die entsprechende Verzeichniseintragung übernommen. Die Datei muß aber vorher durch die Funktion #15 oder #22 aktiviert worden sein. Wurde von der Datei nur gelesen, ist kein Schließen erforderlich, aber nach Schreiboperationen ist die Funktion zur Aktualisierung der Verzeichniseintragung auf der Diskette unbedingt erforderlich.

Steht nach dem Funktionsaufruf im CPU-Register A ein Wert zwischen 00H und 03H, so war die Funktion erfolgreich. Wurde der Name der Datei nicht gefunden, so enthält CPU-Register A den Wert 0FFH.

Funktion 17: Suchen nach erster Eintragung

Aufrufparameter:

Register C: 11H
Register DE: FCB-Adresse

Rückgabewert:

Register A: Verzeichniscode

Mit dieser Funktion werden dem Benutzer (vor allem dem Systemprogrammierer) die Verzeichniseintragen zugänglich gemacht. Das Verzeichnis wird von Anfang an nach einer Eintragung entsprechend den FCB-Eintragen

dr, f1...f8, t1...t3, ex

durchsucht.

Das Registerpaar DE enthält wieder die FCB-Adresse. Fragezeichen im Namen lassen an diesen Positionen jeden Buchstaben zu. Ein Fragezeichen im Byte 12 des FCB (ex) ermöglicht das Suchen aller Einträge mit dem angegebenen Namen. Ein Fragezeichen im Byte 0 des FCB (dr) ermöglicht das Durchsuchen aller Eintragungen, auch der gelöschten, und aller USER-Bereiche im aktuellen Laufwerk.

Ein erfolgreiches Suchen wird im Register A durch die Werte 00H, 01H, 02H und 03H angezeigt. Im anderen Fall enthält das Register den Wert 0FFH.

Die Eintragung, die entsprechend den FCB-Vorgaben ermittelt wurde und 32 Bytes lang ist, wird in einem 128-Byte-Puffer eingetragen. Dieser beginnt entweder bei 0080H oder bei einer mit der Funktion 26 vorgegebenen Adresse. Je nach dem Wert von A ist die Eintragung ab dem ersten Byte (A=00H), 33. Byte (A=01H), 65. Byte (A=02H) oder 97. Byte (A=03H) innerhalb des Puffers abgelegt.

Funktion 18: Suchen nach der nächsten Eintragung

Aufrufparameter:

Register C: 12H

Rückgabewert:

Register A: Verzeichniscode

Diese Funktion wirkt ähnlich der Funktion #17. Das Durchsuchen des Verzeichnisses erfolgt aber ab der aktuellen Position. Diese Funktion ist im Anschluß an die Funktion #17 zu programmieren, wenn nach weiteren Eintragungen der gleichen Datei (große Dateien können mehrere Eintragungen belegen) gesucht wird. Dabei ist in Byte 12 des FCB (ex) die Nummer des gesuchten Bereiches oder 3FH (?) einzutragen.

Funktion 19: Löschen einer Datei

Aufrufparameter:

Register C: 13H

Register DE: FCB-Adresse

Rückgabewert:

Register A: Verzeichniscode

Im Registerpaar DE ist die FCB-Adresse vorzugeben. Mit dieser Funktion wird die Datei auf der Diskette gelöscht. Enthält der Dateiname oder der Dateityp im FCB ein oder mehrere Fragezeichen, so werden alle Dateien gelöscht, die in den anderen Positionen gleiche Zeichen enthalten.

Steht als Rückmeldung im Register A 0FFH, so wurde keine Datei auf der Diskette gefunden, die der Dateibezeichnung im FCB entspricht.

Erfolgreiches Löschen einer Datei wird mit den Werten 00H, 01H, 02H oder 03H im Register A zurückgemeldet.

Es werden alle zu den angegebenen Dateien gefundenen Einträge gelöscht.

Funktion 15: Umbenennen einer Datei

Aufrufparameter:

Register C: 17H

Register DE: FCB-Adresse

Rückgabewert:

Register A: Verzeichniscode

Mit dieser Funktion kann auf der Diskette eine Datei umbenannt werden.

Das Registerpaar DE adressiert einen FCB, der etwas anders aufgebaut ist als bei den anderen Funktionen:

Die Bytes 0 bis 15 enthalten im üblichen Format die "alte" Dateibezeichnung einschließlich Laufwerksangabe, die Bytes 16 bis 32 die "neue" Dateibezeichnung. Hier ist die Laufwerksangabe bedeutungslos. Im CPU-Register A steht nach erfolgreicher Ausführung ein Wert zwischen 00H und 03H. Wenn die umzubenennende Datei nicht gefunden wurde, so steht im CPU-Register der Wert 0FFH. Es wird nicht überprüft, ob vorher schon eine Datei mit der "neuen" Bezeichnung vorhanden ist. Namensdoppelungen sind also möglich.

9.4.3. Adressierung und Datentransport

Für den Datentransport sind folgende Parametereinstellungen notwendig:

- Adressierung des Speicherbereiches für den 128-Byte-Puffer (DMA-Adressierung)
- Adressierung des zu transportierenden Satzes
- Richtung des Transportes (Lesen/Schreiben)

Die Adressierung des zu transportierenden Satzes wird entweder dadurch vorgenommen, daß eine beliebige Satznummer vorgewählt und in den erweiterten FCB eingetragen wird (wahlfreier Zugriff), oder die Satznummer von BDOS selbst verwaltet wird (sequentieller Zugriff). Im letzten Fall wird der Zeiger im FCB, der die aktuelle Satznummer beinhaltet, bei sequentiellm Zugriff auf den nachfolgenden Satz der Datei gestellt.

Durch Eintragungen der Satznummer für sequentiellen Zugriff kann der Satz bestimmt werden, ab dem BDOS die Verwaltung übernehmen soll.

Die DMA-Adresse wird durch die Transportoperationen nicht verändert. Bei direktem Zugriff wird der Satzzeiger nicht geändert, aber der sequentielle Zeiger verweist auf den Zugriffspunkt. Damit ist ein Übergang von direktem zu sequentiellm Zugriff möglich. Bei Zugriffswechsel in umgekehrter Richtung existiert eine zusätzliche Transformationsfunktion. Durch diese wird der Satzzeiger im FCB für direkten Zugriff, entsprechend dem Stand des sequentiellen Zeigers, aktualisiert. Aufgrund der dynamischen Verwaltung der Diskettenbelegung bei SCPX wird auch die Funktion des Schreibens von Sätzen mit Blockinitialisierung angeboten.

Es stehen folgende BDOS-Funktionen zur Verfügung:

Funktion 26: Einstellen der DMA-Adresse

Aufrufparameter:

Register C: 1AH

Register DE: DMA-Adresse

Für den Datentransport von und zur Diskette ist ein Puffer mit 128 Bytes Kapazität erforderlich. Nach Kalt-, Warmstart und Rücksetzen des Diskettensystems (BDOS-Funktion #13) hat dieser Puffer die Adresse 0080H.

Durch diese Funktion kann eine andere Adresse für diesen Puffer eingestellt werden. Sie ist im Registerpaar DE vorzugeben. Diese DMA-Adresse bleibt wirksam bis zur nächsten BDOS-Funktion #26, BDOS-Funktion #13 oder bis zum nächsten Kalt- oder Warmstart.

Funktion 20: Sequentielles Lesen

Aufrufparameter:

Register C: 14H

Register DE: FCB-Adresse

Rückgabewert:

Register A: Fehlercode

Im Registerpaar DE ist die FCB-Adresse vorzugeben. Mit dieser Funktion werden 128 Bytes (ein Satz) von dieser Datei in den Speicher eingelesen. Die Speicheranfangsadresse (DMA-Adresse) ist entweder 0080H oder sie wird durch die BDOS-Funktion #26 vorgegeben. Die Angabe "cr" im FCB gibt den nächsten zu lesenden Satz an (cr = 00H bedeutet also, es wird der 1. Satz gelesen). Nach

dem Lesen wird "cr" automatisch um 1 erhöht. Wird die Extentgrenze erreicht, so wird auch automatisch auf den folgenden Extent umgeschaltet und "cr" auf 00H gesetzt. Wird im Register A ein Wert ungleich 00H gemeldet, so ist das Dateiende erreicht. Bei erfolgreichem Lesevorgang steht im CPU-Register A der Wert 00H.

Funktion 21: Sequentielles Schreiben

Aufrufparameter:

Register C: 15H
Register DE: FCB-Adresse

Rückgabewert:

Register A: Fehlercode

Im Registerpaar DE ist die FCB-Adresse vorzugeben. Von der aktuellen DMA-Adresse wird ein Satz in die durch den FCB vorgegebene Datei übertragen, wobei "cr" die Position des Satzes in der Datei festlegt. Nach dem Schreiben wird "cr" automatisch um 1 erhöht. Das Umschalten auf einen nächsten Extent erfolgt ebenfalls automatisch, "cr" wird dabei auf 00H gesetzt.

Ein erfolgreicher Schreibversuch wird mit 00H im CPU-Register A zurückgemeldet. Ist die Diskette gefüllt und der Satz kann nicht mehr aufgezeichnet werden, so entsteht im CPU-Register A ein Wert ungleich 00H.

Beachte: Nach dem letzten Schreiben muß die CLOSE-Funktion folgen, damit die Informationen des Verzeichnisses auf der Diskette aktualisiert werden.

Lese- und Schreibfehler führen zu der Konsolmeldung:

SCPX ERR ON d: BAD SECTOR (d gibt das logische Laufwerk an: A, B, ...P).

Mit der Abschlußtaste wird die Fehlermeldung ignoriert und das Programm fortgesetzt. Dabei steht im CPU-Register A ein Wert ungleich 00H!

Es besteht außerdem die Möglichkeit, mit ^C das Programm durch Sprung zum Warmstart zu beenden.

Funktion 33: Lesen mit direktem Zugriff

Aufrufparameter:

Register C: 21H
Register DE: FCB-Adresse

Rückgabewert:

Register A: Direktzugriffscod

Für den direkten Zugriff sind die Bytes r0, r1 und r2 erforderlich, dessen Adresse im Registerpaar DE stehen muß. In r0 ist der L-Teil und in r1 der H-Teil der Adresse des Satzes einzutragen, der gelesen werden soll. Das Byte r2 ist auf 00H zu setzen.

Bei den vorher erforderlichen BDOS-Funktionen #15 bzw. #22 (OPEN, Anlegen Datei) ist zu garantieren, daß im adressierten FCB das Byte 12 (ex) auf 00H gesetzt ist, da sonst die erforderliche Synchronisation zwischen Extentnummer und Satzadresse vom BDOS nicht hergestellt werden kann.

Der Satz wird im durch die DMA-Adresse vorgegebenen Puffer eingetragen.

Im Gegensatz zum sequentiellen Lesen wird die Satzadresse nach dem Zugriff nicht um 1 erhöht, so daß eine sofort folgende BDOS-Funktion #33 den gleichen Satz lesen würde.

Nach dem Lesen werden durch diese Funktion außerdem die aktuelle Satznummer "cr" und die Extentnummer "ex" gesetzt. Damit ist ein anschließendes sequentielles Arbeiten möglich.

Es ist zu beachten, daß der zuletzt gelesene Satz eingestellt ist, d.h., er wird dann nochmals gelesen bzw. überschrieben.

Bei erfolgreichem Lesen wird im Register A eine 00H gemeldet. Da nicht alle Satzadressen beim direkten Zugriff zwischen niedrigstem und höchstem Wert existieren müssen (siehe BDOS-Funktion #34), kann es zu folgenden Fehlermeldungen im CPU-Register A kommen:

- 01H Es sollte ein Satz in einem nicht beschriebenen Block gelesen werden. Das Lesen von nicht geschriebenen Sätzen in vorhandenen Blöcken erzeugt keine Fehlermeldung.
- 03H Vor diesem Funktionsaufruf wurde die Funktion Schreiben mit direktem Zugriff ausgeführt. Das jetzt eventuell erforderliche Aktualisieren der Verzeichniseintragung, durch das vorherige Schreiben hervorgerufen, konnte nicht ausgeführt werden.
- 04H Es sollte ein Satz von einem unbeschriebenen Extent gelesen werden.
- 06H Mit der eingestellten Satzadresse würde das Diskettenende überschritten werden.

Funktion 34: Schreiben mit direktem Zugriff

Aufrufparameter:

Register C: 22H
Register DE: FCB-Adresse

Rückgabewert:

Register A: Direktzugriffscod

Wie beim Lesen mit direktem Zugriff geben das Registerpaar DE die FCB-Adresse und dort die Bytes r0 und r1 die Adresse des Satzes an. An diese Adresse wird der Satz aus dem durch die DMA-Adresse vorgegebenen Puffer geschrieben und, falls ein neuer Extent dafür erforderlich ist, dieser Extent in das Verzeichnis eingetragen.

Die Satzadresse wird wie beim Lesen nicht erhöht.

Es werden aber die aktuelle Satznummer "cr" und die Extentnummer "ex" im FCB aktualisiert, so daß ein anschließendes sequentielles Arbeiten (wie beim Lesen mit direktem Zugriff beschrieben) möglich ist.

Hinweis: Sobald ein Satz in einen SCP-Diskettenblock der Datei (vgl. Abschnitt 11.6.2) geschrieben wurde, gelten alle anderen Sätze in diesem Block als geschriebene Daten. Das Lesen eines dieser Blöcke bringt also keinen Fehlercode 01H.

Die Fehlercodes sind die gleichen wie beim Lesen mit direktem Zugriff (außer 01H und 04H), dazu kommt der Fehlercode 05H. Diese Fehlermeldung signalisiert, daß in dem Verzeichnis für die erforderliche neue Eintragung kein Platz mehr frei ist.

Funktion 40: Schreiben mit direktem Zugriff und Blockinitialisierung

Aufrufparameter:

Register C: 28H
Register DE: FCB-Adresse

Rückgabewert:

Register A: Direktzugriffscod

Diese Funktion entspricht der BDOS-Funktion #34 mit der Erweiterung, daß ein neuangelegter Block automatisch initialisiert wird. Alle Sätze des Blockes werden mit dem Wert 0 vorbelegt.

Funktion 36: Bereitstellen aktuelle Satzposition

Aufrufparameter:

Register C: 24H
Register DE: FCB-Adresse

Rückgabewert:

Satzadressen des letzten sequentiellen Zugriffs

Das Registerpaar DE adressiert den FCB, in dessen Bytes r0 und r1 durch diese Funktion die Satzadressen des letzten sequentiellen Zugriffs abgelegt werden. Damit kann z.B. auf einfache Weise vom sequentiellen zum direkten Zugriff übergegangen werden.

9.4.4. System- und Hilfsfunktionen

Mit diesen Funktionen werden die Initialisierungen des Betriebssystems ausgeführt bzw. die Abfrage und Beeinflussung von Betriebssystemzuständen ermöglicht.

Zu unterteilen ist in

- Systemfunktionen
- Laufwerksfunktionen
- Dateiverwaltungsfunktionen.

Systemfunktionen

Funktion 0: System-Warmstart

Aufrufparameter:

Register C: 00H

Mit dieser Funktion erfolgt die Rückkehr in den CCP-Status durch einen Warmstart. Die Wirkung entspricht der Funktion Warmstart des BIOS-Sprungverteilers. Die BDOS-Funktion verzweigt dorthin. Auf der absoluten Speicheradresse 0000H ist ein Sprung in den BIOS-Warmstart eingetragen.

Zur Adresse 0000H verzweigen die Eingabe von ^C und BDOS-Fehler.

Beim Warmstart wird die DMA-Adresse 80H eingestellt und anschließend die Systemsteuerung an das CCP übergeben.

Funktion 12: Abfrage der Versionsnummer

Aufrufparameter:

Register C: 0CH

Rückgabewert:

Register HL: Versionsnummer

Mit dieser Funktion kann das Nutzerprogramm die Leistungsfähigkeit des SCPX abfragen. Als Antwort erhält das aufrufende Programm im CPU-Doppelregister HL den Wert 0022H. Das symbolisiert eine Leistungsfähigkeit, die der vom CP/M 2.2 entspricht.

Funktion 32: Einstellen/Abfragen Benutzercode

Aufrufparameter:

Register C: 20H

Register E: 0FFH (Abfragen) oder Benutzercode (Einstellen)

Rückgabewert:

Register A: aktueller Benutzercode oder kein Wert

Wird im CPU-Register E ein 0FFH eingetragen, so wird nach dem Funktionsaufruf im CPU-Register A der aktuelle Benutzercode zurückgemeldet.

Ist der Wert im Register ungleich 0FFH, so werden durch die Funktion #32 die unteren 5 Bits dieses Wertes als neuer Benutzercode eingestellt (00H...1FH).

Beachte: Mit dem residenten Kommando USER lassen sich nur die Benutzerbereiche 0 bis 15 (00H...0FH) einstellen, so daß die anderen Kommandos nur in diesen Benutzerbereichen wirksam werden können. Es sollte deshalb generell nur mit dem Benutzercode 00H...0FH gearbeitet werden.

Funktion 31: Bereitstellen der Adresse der Laufwerksparameter

Aufrufparameter:

Register C: 1FH

Rückgabewert:

Register HL: DPB-Adresse

Nach dem Funktionsaufruf steht im CPU-Registerpaar HL die Adresse der Parametertabelle des aktuellen Laufwerkes. Diese Tabelle ist im BIOS-Teil angelegt und enthält Organisationsparameter des entsprechenden Laufwerkes, die auch von BDOS herangezogen werden. In diesem Parameterfeld kann der Systemprogrammierer spezielle Informationen auswerten oder Manipulationen der Parameter durchführen.

Funktion 27: Bereitstellen der Adresse des Allocationsvektors

Aufrufparameter:

Register C: 1BH

Rückgabewert:

Register HL: Adresse des Allocationsvektors

Für jedes logische Laufwerk, auf das einmal zugegriffen wurde, gibt es im Speicher eine Tabelle, die die Belegung der Diskette (belegte Blöcke - freie Blöcke) widerspiegelt.

Mit dieser Funktion wird im Registerpaar HL die Adresse dieser Tabelle (Allocationvektor) für das aktuelle Laufwerk bereitgestellt.

Die Tabelle beinhaltet auch das Verzeichnis, aber nicht die Systemspuren der Diskette.

Diese Funktion ist normalerweise nur für den Systemprogrammierer von Bedeutung.

Laufwerksfunktionen

Im SCPX sind 16 logische Laufwerke A bis P definiert. Die BDOS-Funktionen arbeiten mit diesen 16 Laufwerken. Dabei ist zu gewährleisten, daß im BIOS alle die Laufwerke installiert sind, die vom BDOS-Aufrufer angefordert werden.

Im BIOS existiert ein ausgezeichnetes Laufwerk, das Laufwerk A, und ein aktuelles Laufwerk, das selektiert wird und bis zur Selektion eines anderen Laufwerkes selektiert bleibt. Dieses Laufwerk hat eine bevorzugte Stellung; die Parameterübergabe für andere BDOS-Funktionen bezieht sich auf diese Selektion. Jedes Laufwerk, auf das nach einer Initialisierung das erste Mal zugegriffen wird, erhält den "on-line-Zustand", alle anderen bleiben im "off-line-Zustand". Jedes Laufwerk kann durch Setzen des entsprechenden Schreibschutzes in den "read-only-Zustand" versetzt werden.

Funktion 13: Rücksetzen des Diskettensystems

Aufrufparameter:

Register C: 0DH

Rückgabewert:

Register HL: Batchmode-Flag

Um in einem Programm Disketten ohne Warmstart (der zum Programmabbruch führt) wechseln zu können, ist diese Funktion erforderlich. Sonst führt ein Diskettenwechsel ohne Warmstart oder ohne anschließende BDOS-Funktion #13 zu R/O-Fehler. Mit dieser Funktion wird das LW A das aktuelle Laufwerk, alle Laufwerke erhalten den R/W-Status (siehe BDOS-Funktion #28), die aktuelle DMA-Adresse wird auf 0080H eingestellt (siehe BDOS-Funktion #26), und alle Laufwerke außer LW A erhalten den "off-line-Zustand".

Wird im Register A=00H zurückgegeben, so ist auf der Diskette keine Batchmode-Datei vorhanden, bei A=FFH ist sie vorhanden. Der Name der Batchmode-Datei beginnt mit "\$" (f1=3FH), der Typ ist "SUB".

Funktion 37: Laufwerke zurücksetzen

Aufrufparameter:

Register C: 25H
Register DE: Reset-Marke

Rückgabewert:

Register A: 00H

Mit dieser Funktion können Laufwerke einzeln zurückgesetzt werden (im Gegensatz zu BDOS-Funktion #13, die alle logischen Laufwerke zurücksetzt). Damit können einzelne Laufwerke zurückgesetzt werden, um entweder Disketten in den rückgesetzten Laufwerken wechseln zu können oder den Schreibschutz für diese Laufwerke zu löschen. Im CPU-Doppelregister DE wird die Reset-Marke eingetragen.

Jedem der 16 Bits ist ein logisches Laufwerk A,...,P in folgender Art zugeordnet:

Bit 0, E : LW A

Bit 1, E : LW B

.....

Bit 7, D : LW P

Die 1-Belegung kennzeichnet die zurückgesetzten Laufwerke.

Funktion 14: Laufwerk selektieren

Aufrufparameter:

Register C: 0EH
Register E: Laufwerk

Der Inhalt vom Register E gibt an, welches logische Laufwerk für die folgenden Dateioperationen das aktuelle Laufwerk ist.

Dabei gilt die folgende Zuordnung:

E=00 : LW A

E=01 : LW B

.....

E=15 : LW P

Ist z.B. mit dieser Funktion das LW C selektiert, so bedeutet der Wert 00 im ersten Byte des FCB, daß mit diesem FCB bei Dateizugriff die Diskette im Laufwerk C angesprochen wird. Das ausgewählte Laufwerk bleibt jetzt aktiviert (im "on-line-Zustand") bis zum nächsten Rücksetzen des Diskettensystems oder Warmstart. Bei einem Diskettenwechsel während des "on-line-Zustandes" würde das Laufwerk in den R/O-Status gehen.

Funktion 25: Abfrage nach aktuellem Laufwerk

Aufrufparameter:
Register C: 19H

Rückgabewert:
Register A: aktuelles Laufwerk

Diese BDOS-Funktion meldet im CPU-Register A das aktuelle logische Laufwerk.

Es gilt die Zuordnung:

A=00 : LW A

A=01 : LW B usw.

Funktion 29: Abfrage nach Laufwerk mit Schreibschutz

Aufrufparameter:
Register C: 1DH

Rückgabewert:
Register HL: Schreibschutzwert
A=L

In den CPU-Registern H und L wird wie bei BDOS-Funktion #24 ein Code abgelegt, aus dem zu entnehmen ist, welches logische Laufwerk den R/O-("read-only")-Status hat. Das diesen Laufwerken zugeordnete Bit ist auf "1" gesetzt. Der R/O-Status kann durch die BDOS-Funktion #28 und durch einen Diskettenwechsel (ohne folgende BDOS-Funktion #13) gesetzt sein.

Funktion 24: Abfrage nach Laufwerk im "on-line-Zustand"

Aufrufparameter:
Register C: 18H

Rückgabewert:
Register HL: Zustandswert
A=L

Mit dieser BDOS-Funktion läßt sich feststellen, mit welchen Laufwerken seit der letzten Systeminitialisierung (Kaltstart, Warmstart, Reset des Disketten-Systems) gearbeitet wurde. In den CPU-Registern H und L wird dieses zur Verfügung gestellt.

In diesem 16-Bit-Vektor wird jedes Bit einem LW folgendermaßen zugeordnet:

Bit 0, L : LW A

Bit 1, E : LW B

.....

Bit 7, L : LW H

Bit 0, H : LW I

Bit 1, H : LW J

.....

Bit 1, H : LW P

Der Bitwert 0 stellt den "off-line-Zustand", der Bitwert 1 stellt den "on-line-Zustand" dar.

Dateiverwaltung

Funktion 30: Setzen der Dateiattribute

Aufrufparameter:

Register C: 1EH

Register DE: FCB-Adresse

Rückgabewert:

Register A: Verzeichniscode

Mit dieser Funktion können Dateien mit den Attributen R/O (Nur-Lese-Datei) oder R/W (Lese-Schreib-Datei) und DIR (Datei wird beim residenten Kommando DIR angezeigt) oder SYS (keine Anzeige bei DIR) versehen werden (siehe auch Abschnitt 3.7).

Das CPU-Registerpaar DE adressiert einen FCB. Ist das Bit 7 von t1 gesetzt, so wird die Datei mit dieser Funktion zu einer R/O-Datei. Ist das Bit nicht gesetzt, wird die Datei zu einer R/W-Datei. Ein gesetztes Bit 7 von t2 versetzt die Datei in den SYS-Datei-Status. Im anderen Fall erhält die Datei das DIR-Attribut.

Funktion 35: Bereitstellen der nächsten freien Satzposition

Aufrufparameter:

Register C: 23H

Register DE: FCB-Adresse

Rückgabewert:

Adresse nach letztem Satz

Mit dem CPU-Registerpaar DE ist ein FCB zu adressieren. Nach Ausführung der Funktion enthalten die Bytes r0 und r1 die Adresse nach dem letzten Satz der Datei. Damit kann die (virtuelle) Größe einer Datei bestimmt werden. Nach der Funktion #35 ist es einfach möglich, eine Datei mit Direktzugriff weiterzuschreiben.

10. Ein Beispiel für die Anwendung der BDOS-Funktionen

Mit dem folgenden einfachen Programmbeispiel soll die prinzipielle Anwendung der BDOS-Funktionen demonstriert werden.

Aufgabe dieses Erfassungsprogramms ist:

- Der Bediener gibt den Dateinamen vor (Dateityp ist TEX). Eine eventuell bestehende Datei gleicher Bezeichnung wird gelöscht!
- Dann kann er Sätze bis zu max. 126 Zeichen eingeben, die auf dem Bildschirm dargestellt werden. Gibt er weniger als 126 Zeichen ein, so hat er mit der Abschlußtaste die Eingabe zu beenden. Korrekturen vor Abschluß eines Satzes sind möglich (entsprechend BDOS-Funktion 10).
- Jeder Satz, am Ende mit Leerzeichen aufgefüllt und mit den Steuerzeichen Zeilenumschaltung und Cursor an Zeilenanfang versehen, wird auf der Diskette im aktuellen Laufwerk als ein Satz mit 128 Bytes in der vorgewählten Datei aufgezeichnet.
- Gibt der Bediener nur "END" bzw. "end" ein, so wird dies als letzter Satz gedeutet. Das Programm wird nach dem Aufzeichnen dieses Satzes und dem Schließen der Datei beendet.
- Tritt ein Fehler auf, so wird die Datei nicht aufgezeichnet. Der Fehler wird angezeigt.

```

beispiel1      ASM 1520      (SCPX)      V 1.0      1
                                                    title      beispiel1

0000          aseq
              org      100h
              .z80
              page     63

0005          bdos      equ      5          ;Adresse vom Sprung an BDOS
000D          cr       equ      0dh        ;Steuerzeichen:
                                                    ;Cursor an Zeilenanfang
000A          lf       equ      0ah        ;Steuerzeichen:
                                                    ;Zeilenschaltung
000C          home    equ      0ch        ;Steuerzeichen:
                                                    ;Cursor an BS-Anf.

0100      31 0269  start:   ld      sp,stack+16    ;eigener Stack
0103      11 0269          ld      de,text          ;Programmanzeige
                                                    ;auf Bildschirm

0106      0E 09          ld      c,9
0108      CD 0005        call   bdos          ;*** BDOS-Ruf 9 ***

010B      11 0283          ld      de,name          ;Dateiname:
                                                    ;Eingabeaufforderung
010E      0E 09          ld      c,9
0110      CD 0005        call   bdos          ;*** BDOS-Ruf 9 ***
0113      11 01B6        ld      de,puffer          ;Eingabe Dateiname

0116      0E 0A          ld      c,10
0118      CD 0005        call   bdos          ;*** BDOS-Ruf 10 ***
011B      3A 01B7        ld      a,(puffer+1)    ;Anzahl Zeichen
                                                    ;Dateiname

011E      4F            ld      c,a
011F      06 00          ld      b,0
0121      21 01B8        ld      hl,puffer+2
0124      11 0239        ld      de,fcbl+1
0127      ED B0          ldir          ,Dateiname--->FCB

0129      3E /E          ld      a,126          ;max. Eing.länge
                                                    ;f.Satzeingabe
012B      32 01B6        ld      (puffer),a
012E      11 0238        ld      de,fcbl          ;Löschen einer evtl.
                                                    ;bestehenden Datei
0131      D5            push   de          ,gleichen Namens

```

```

beispiel1      ASM 1520      (SCPX)      V 1.0      1

0132      0E 13      ld      c,19
0134      CD 0005      call    bdos      ;*** BDOS-Ruf 19 ***

0137      D1      pop    de      ;Einrichten Datei
0138      D5      push   de
0139      0E 16      ld      c,22
013B      CD 0005      call    bdos      ;*** BDOS-Ruf 22 ***

013E      3C      inc    a
013F      28 64      jr     z,fehler   ;Sprung bei Fehler

0141      11 01B8      ld      de,dma    ;Adresse für FD-Übertragung
0144      0E 1A      ld      c,26
0146      CD 0005      call    bdos      ;*** BDOS-Ruf 26 ***
0149      1E 0D      loop:  ld      e,cr   ;Jurnsoreinstellung
014B      0E 02      ld      c,2
014D      CD 0005      call    bdos      ;*** BDOS-Ruf 2 ***

0150      1E 0A      ld      e,lf
0152      0E 02      ld      c,2
0154      CD 0005      call    bdos      ;*** BDOS-Ruf 2 ***

0157      11 01B6      ld      de,puffer ;Texteingabe
015A      0E 0A      ld      c,10
015C      CD 0005      call    bdos      ;*** BDOS-Ruf 10 ***

015F      21 01B8      ld      hl,dma    ;Ausgabepuffer FD
0162      3A 01B7      ld      a,(puffer+1) ;Anzahl Zeichen
0165      4F      ld      c,a
0166      06 00      ld      b,0
0168      09      add    hl,bc      ;erstes zu löschendes Byte
0169      3E 7E      ld      a,126    ;max. Anzahl Zeichen
016B      91      sub    c
016C      28 07      jr     z,write    ;bei 126 Zeichen

016E      47      ld      b,a
016F      3E 20      ld      a,20h    ;füllzeichen
0171      77      loesch: ld      (hl),a
0172      23      inc    hl
0173      10 FC      djnz  loesch
0175      D1      write: pop    de      ;Write sequentiell
0176      D5      push   de
0177      0E 15      ld      c,21
0179      CD 0005      call    bdos      ;*** BDOS-Ruf 21 ***
017C      B7      or     a
017D      20 26      jr     nz,fehler ;Sprung zur Fehlerbehandlung
017F      3A 01B7      ld      a,(puffer+1) ;Abfrage auf "end" bzw. "END"
0182      FE 03      cp     3
0184      20 C3      jr     nz,loop   ;keine 3 Zeichen eingegeben
0186      3A 01B8      ld      a,(puffer+2)
0189      CB AF      res    5,a

018B      FE 45      cp     'E'      ;Großbuchstaben erzeugen
018D      20 BA      jr     nz,loop  ;Fortsetzen der Erfassung
018F      3A 019B      ld      a,(puffer+3)
0192      CB AF      res    5,a
0194      FE 4E      cp     'N'
0196      20 B1      jr     nz,loop  ;Fortsetzen der Erfassung
0198      3A 01BA      ld      a,(puffer+4)
019B      CB AF      res    5,a
019D      FE 44      cp     D
019F      20 AB      jr     nz,loop  ;Fortsetzen der Erfassung
01A1      0E 10      ld      c,16    ;Schließen der Datei
01A3      18 0A      jr     ende

01A5      11 0291      fehler: ld      de,fetext ;fehleranzeige
01A8      0E 09      ld      c,9
01AA      CD 0005      call    bdos      ;*** BDOS-Ruf 9 ***

01AD      0E 13      ld      c,19    Löschen Datei im Fehlerfall
01AF      D1      pop    de      ;FCB-Adresse
01B0      CD 0005      call    bdos      ;*** BDOS-Ruf 19 ***
01B3      C3 0000      jp     0        ;Ende mit Warmstart

01B6      08 00      puffer: db    B,0 ;Eingabepuffer
                                ;(max. Länge, Istlänge)
01B8      dma:      ds    126     ;Puffer für Eingabe
                                ;und FD-Ausgabe
0236      0D 0A      db    cr,lf    ;letzte konstante Zeichen f.

0238      00      fcb:      db    0        ;FCB:aktuelles Laufwerk
0239      ds    8,20h ; für Dateiname
0241      54 45 58  db    'TEX'   ; Dateityp
0244      ds    21,0  ; Restbyte (auf 0 gestellt)

```

```

beispiel1      ASM 1520      (SCPX)      U 1.0      1

0259           stack:      ds      16,0           ;Anwenderstack

0269 0C 20 20 20 text:      db      home,'      ERFASSUNGSPROGRAMM',
026D 20 45 52 46          cr,lf,24h
0271 41 53 53 55
0275 4E 47 53 50
0279 52 4F 47 52
027D 41 4D 4D 0D
0281 20 24
0283 0A 44 61 74 name:      db      lf,'Dateiname : ',24h
0287 65 69 6E 61
028B 6D 65 20 3A
028F 20 24
0291 0D 0A 0A 20 fetext:   db      cr,lf,lf,'Fehler (Directory oder Diskette
0295 46 65 68 6C          voll) !'
0299 65 72 20 28

029D 44 65 68 6C
02A1 63 74 6F 72
02A5 79 20 6F 64
02A9 65 72 20 44
02AD 69 73 6B 65
02B1 74 74 65 20
02B5 76 6F 6C 6C
02B9 29 20 21
02BC 0D 0A 24           db      cr,lf,24h

                                end

```

Macros:

Symbols:

0005	BDOS	000D	CR	01B8	DMA
01AF	ENDE	0238	FCB	01A5	FEHLER
0291	FETEXT	000C	HOME	000A	LF
0171	LOESCH	0149	LOOP	0283	NAME
01B6	PUFFER	0259	STACK	0100	START
0269	TEXT	0175	WRITE		

No Fatal error(s)

11. Beschreibung der BIOS-Funktionen

11.1. Aufgabe des BIOS

Das BIOS ist der von der Hardware abhängige Teil des Betriebssystems SCPX 5105, der physische Funktionen realisiert. Er beinhaltet die für die spezielle Hardware des Bildungscomputers notwendigen Ein-/Ausgaberoutinen (Gerätetreiber). Damit bildet er die Schnittstelle zwischen der Hardware und dem hardwareunabhängigen Teil des Betriebssystems bzw. dem Anwenderprogramm.

Die im BIOS enthaltenen Ein-/Ausgaberoutinen können in drei Gruppen zusammengefaßt werden:

1. Systeminitialisierung
2. Zeichenein- und -ausgabe
3. Diskettenzugriffsoperationen

Die Routinen erreicht man über einen sogenannten "Sprungvektor". Der Sprungvektor steht am BIOS-Anfang und stellt eine zusammenhängende Folge von Sprungbefehlen dar. Nachfolgend sind die Routinen der oben genannten Gruppen angegeben, zu denen je ein Sprungbefehl im Sprungvektor enthalten ist:

Systeminitialisierung:

- Kaltstartroutine
- Warmstartroutine

Zeichenein-/ausgabeoperationen:

- Status CONSOLE-Gerät Eingabe
- Eingabe von CONSOLE-Gerät
- Status CONSOLE-Gerät Ausgabe
- Ausgabe auf CONSOLE-Gerät
- Ausgabe auf LIST-Gerät
- Status LIST-Gerät
- Ausgabe auf PUNCH-Gerät
- Status PUNCH-Gerät
- Eingabe von READER-Gerät
- Status von READER-Gerät
- Tabellenverwaltung der Treiber/Slottabelle

Diskettenzugriffsoperationen:

- Auf der Spur Null positionieren
- Laufwerk auswählen
- Spur auswählen
- Sektornummer transformieren
- Sektor auswählen
- Datenpufferadresse setzen
- Selektierten Sektor lesen
- Selektierten Sektor schreiben

11.2. Schnittstellen des BIOS

Im Betriebssystem SCPX 5105 hat der BIOS-Sprungvektor nachfolgend beschriebenen Aufbau. Die symbolischen Sprungadressen dienen nur zum besseren Verständnis der nachfolgenden Beschreibungen.

Sprungnummer	Befehl	Funktion
0	JP BOOT	; Kaltstartroutine
1	JP WBOOT	; Warmstartroutine
2	JP CONST	; CONSOLE-Status Eingabe abfragen
3	JP CONIN	; CONSOLE-Eingabe
4	JP CONOUT	; CONSOLE-Ausgabe
5	JP LIST	; LIST-Ausgabe
6	JP PUNCH	; PUNCH-Ausgabe
7	JP READER	; READER-Eingabe
8	JP HOME	; Spur Null einstellen
9	JP SELDSK	; Laufwerk auswählen
10	JP SETTRK	; Spur auswählen
11	JP SETSEC	; Sektor auswählen
12	JP SETDMA	; Datenpufferadresse setzen
13	JP READ	; Selektierten Sektor lesen
14	JP WRITE	; Selektierten Sektor schreiben
15	JP LISTST	; LIST-Status abfragen
16	JP SECTRAN	; Sektornummer umrechnen
17	JP CONOST	; CONSOLE-Status Ausgabe abfragen
18	JP READST	; READER-Status abfragen
19	JP PUNST	; PUNCHER-Status abfragen
20	JP DEVTBL	; Treiber/Slottabellenverteiler

Die angegebenen Routinen werden als Unterprogramm aufgerufen, enden also mit einem Rücksprung (mit Ausnahme der Warm- und Kaltstartroutine, für die eigene Regeln gelten). Dabei werden eventuell benötigte Werte in folgenden Prozessorregistern übergeben:

8-Bit-Werte	- an BIOS	in Register C	- vom BIOS	in Register A
16-Bit-Werte		in Registerpaar BC		in Registerpaar HL
zweiter 16-Bit-Wert		in Registerpaar DE		in Registerpaar DE

Ein Programm kann die BIOS-Routinen auch unmittelbar nutzen. Der Ausgangspunkt dazu ist der Sprung auf Adresse 0. Hier befindet sich ein Sprung zur Warmstartroutine im BIOS-Teil (zweite Eintragung im Sprungvektor). Aus der Zieladresse dieses Sprungbefehls und der Nummer der benötigten BIOS-Routine läßt sich leicht die Adresse berechnen, die das Programm gegebenenfalls aufrufen muß:

$$(\text{Sprungnummer} - 1) * 3 + \text{Zieladresse_des_Sprungs_auf_Adresse } 0$$

Wenn ein Programm beispielsweise den Zustand des CONSOLE-Gerätes (Sprungnummer = 2) wissen möchte, dann kann das mit dem folgenden Unterprogramm geschehen:

```
LD    DE, (2-1)*13    ; ((Nummer der BIOS-Routine) - 1)*3
LD    HL, (1)         ; Adresse Warmstart
ADD   HL, DE          ; Adresse der BIOS-Routine
JP    (HL)            ; Sprung zur BIOS-Routine
```

Hinweis: Dieses Unterprogramm kann nicht zur Berechnung der Adresse der Kaltstartroutine verwendet werden !

11.3. Initialisieren des SCPX 5105 (BOOT, WBOOT)

Es gibt im BIOS zwei Routinen zur Initialisierung des SCPX 5105, die Kaltstart- und die Warmstartroutine.

Die Kaltstartroutine BOOT wird nach dem Einschalten des Rechners und dem Vorfinden der Systemdatei SCPX5105.SYS automatisch aktiviert oder nach dem direkten Aufruf der BIOS-Funktion BOOT. Sie führt eine grundlegende Systeminitialisierung, auch der Hardware, durch und gibt den

Systemkaltstarttext

```
BIC System
Version 2.00

robotron MKD 1989
```

auf dem Bildschirm aus. Zusätzlich wird versucht, das Kaltstartkommando SUBM AUTOEXEC.SUB auszuführen. Das gelingt, wenn die Dateien SUBM.COM und AUTOEXEC.SUB auf der eingelegten Diskette vorhanden sind. Dabei kann die Datei AUTOEXEC.SUB leer sein (vgl. Abschnitt 4.2.5). Sind eine bzw. beide Dateien nicht vorhanden, so erscheinen die entsprechenden Mitteilungen

```
SUBM?
oder
```

```
--Error On Line 001 No ,SUB' File Present
```

auf dem Bildschirm. Die Initialisierung schließt mit der Übergabe der Steuerung an das CCP ab. Das System geht in den Grundzustand über.

Die Warmstartroutine WBOOT wird immer dann aktiviert, wenn ein Nutzerprogramm zur Adresse 0000H verzweigt, die BDOS-Funktion #0 oder direkt die BIOS-Funktion WBOOT aufgerufen wird (vgl. Abschnitt 9.4.4). Nach der Initialisierung der Systemparameter wird zum CCP verzweigt.

11.4. Zeichenweiser Datenverkehr über die logischen Geräte (CONST, CONIN, CONOUT, CONOST, LIST, LISTST, PUNCH, PUNST, READER, READST, DEVTBL)

11.4.1. Logische Ein-/Ausgabekanäle

Das System SCPX unterstützt 4 Kanäle:

- einen Ein-/Ausgabekanal
- einen Eingabekanal
- zwei Ausgabekanäle

Über die im Abschnitt 9 beschriebenen BDOS-Funktionen sind diese Kanäle ansprechbar.

Die Unterprogramme, höhere Programmiersprachen und verschiedene Kommandos (z.B. STAT, PIP usw. siehe dazu Abschnitt 4.2.3 und 4.2.4) greifen über die BDOS-Funktionen auf die E/A-Kanäle zu.

Die Schlüsselbegriffe CON:, LST:, PUN: und RDR: stellen die Kanalbezeichnungen dar, wie sie u.a. in den Programmen PIP und STAT Verwendung finden. Die 4 logischen E/A-Kanäle haben folgende Eigenschaften:

CON: (CONSOLE)
Dieser logische Kanal kann sowohl zur Ein- als auch zur Ausgabe von Zeichen genutzt werden. Im Normalfall sind Tastatur (Eingabe) und Bildschirm (Ausgabe) zugewiesen.

Eingaberoutinen:
CONIN für zeichenweise Eingabe
CONST zur Abfrage, ob ein Zeichen bereitsteht

Ausgaberoutinen:
CONOUT für zeichenweise Ausgabe
CONOST zur Abfrage, ob Ausgabe erfolgt ist

LST: (LIST)
Der logische Kanal dient zur Ausgabe einzelner Zeichen.
Das LIST-Gerät ist normalerweise ein Drucker.

Eingaberoutinen:
LISTST zur Statusabfrage des LIST-Gerätes

Ausgaberoutinen:
LIST für zeichenweise Ausgabe

PUN: (PUNCH)
Der logische Kanal dient zur zeichenweisen Ausgabe.

Eingaberoutinen:
PUNST für Statusabfrage des PUNCH-Gerätes

Ausgaberoutinen:
PUNCH für zeichenweisen Ausgabe

RDR: (READER)
Der logische Kanal dient zur zeichenweisen Eingabe

Eingaberoutinen:
READER für zeichenweise Eingabe
REAST zur Statusabfrage des READER-Gerätes

Jedem logischen E/A-Kanal kann genau einer von jeweils 4 möglichen Subkanälen zugeordnet werden. Diese Zuordnung wird durch den Inhalt des I/O-Bytes auf Adresse 0003H bestimmt. Die durch das I/O-Byte festgelegte Zuordnung des logischen Kanals zum Subkanal kann durch BDOS-Funktionen abgefragt und/oder geändert werden (siehe dazu Abschnitt 9.3.1). Auf Kommandoebene kann eine solche Zustandsabfrage bzw. -änderung des I/O-Bytes mit STAT erfolgen (siehe dazu auch Abschnitt 4.2.3).

Jedem Subkanal kann genau ein physischer Gerätetreiber zugeordnet werden.

Der Aufbau des I/O-Bytes ist wie folgt:

		I/O-Byte, Bit-#							
log. E/A-Kanal	Subkanal	7	6	5	4	3	2	1	0
CON:	TTY:	x	x	x	x	x	x	0	0
	CRT:	x						0	1
	BAT:	x						1	0
	UC1:	x						1	1
RDR:	TTY:	x				0	0		x
	PTR:	x				0	1		x
	UR1:	x				1	0		x
	UR2:	x				1	1		x
PUN:	TTY:	x		0	0				x
	PTP:	x		0	1				x
	UP1:	x		1	0				x
	UP2:	x		1	1				x
LST:	TTY:	0	0						x
	CRT:	0	1						x
	LPT:	1	0						x
	UL1:	1	1	x	x	x	x	x	x

Die Subkanäle gleichen Namens bei verschiedenen logischen E/A-Kanälen können unterschiedliche Bedeutung haben. So können z.B. unter TTY: bei CON:, RDR:, PUN: und LST: völlig unterschiedliche Gerätetreiber angesprochen werden, je nachdem, welche Treiber dem jeweiligen TTY: zugeordnet worden sind.

Im SCPX 5105 sind die Subkanäle wie folgt belegt:

CONSOLE:	CRT:	=	Bildschirm/Tastaturtreiber
	UC1:	=	V.24 bidirektional (1)
READER:	TTY:	=	Tastaturtreiber
	PTR:	=	V.24-Empfänger (1)
PUNCHER:	TTY:	=	Bildschirmtreiber
	PTP:	=	V.24-Sender (1)
	UP2:	=	V.24-Sender (2)
LIST:	CRT:	=	Bildschirmtreiber
	LPT:	=	V.24-Sender (2)
	UL1:	=	V.24-Sender (1)

- (1) - über PLOTTER-Steckverbinder an DSE
- (2) - über PRINTER-Steckverbinder an DSE

Alle anderen Subkanäle sind mit Dummy-Treibern belegt.

11.4.2. BIOS-Schnittstellen für diese Geräte

CONST Abfrage des Status von Kanal CON:

Aufrufparameter	:	-
Rückkehrparameter	:	A - Status

Diese Routine untersucht das dem Kanal CON: zugewiesene Gerät und liefert:
im Register A = 0 falls kein Zeichen bereitsteht
A = 0FFH sonst.

CONIN Empfangen von Zeichen von Kanal CON:

Aufrufparameter	:	-
Rückkehrparameter	:	A - Empfangenes Zeichen

Diese Routine liest das nächste Zeichen von CON: in das Register A. Steht kein Zeichen bereit, wird bis zur Zeicheneingabe gewartet.

CONOUT Senden von Zeichen auf Kanal CON:

Aufrufparameter	:	C - Sende-Zeichen
Rückkehrparameter	:	-

Das in Register C bereitgestellte Zeichen wird auf Kanal CON: ausgegeben.

LIST Senden von Zeichen auf Kanal LST:

Aufrufparameter	:	C - Sende-Zeichen
Rückkehrparameter	:	-

Das in Register C bereitgestellte Zeichen wird auf den Kanal LST: ausgegeben.

LISTST Abfrage des Status von Kanal LST:

Aufrufparameter	:	-
Rückkehrparameter	:	A - Status Kanal LST:

Die Routine übermittelt dem Nutzer eine Status-Information über den Kanal LST:
Register A = 0FFH, wenn der Kanal LST: bereit ist, ein Zeichen zu übernehmen
Register A = 0 Sonstiges

PUNCH Senden von Zeichen auf Kanal PUN:

Aufrufparameter	:	C - Sende-Zeichen
Rückkehrparameter	:	-

Die Routine gibt das in Register C bereitgestellte Zeichen auf den Kanal PUN: aus.

3. Adressierung des Sektors, der die Information enthält, mittels der Routinen SECTAN und SETSEC. Das Einstellen der Sektornummer erfolgt in zwei Schritten. Im ersten Schritt erfolgt über die Routine SECTAN die Umwandlung der logischen Sektornummer in die physische Sektornummer. Diese Umwandlung ermöglicht eine Diskettenorganisation, die einen zeitoptimalen Zugriff auf die gewünschte Information gewährleistet. Im zweiten Schritt wird über die Routine SETSEC die Adressierung des physischen Sektors vorgenommen.

Hinweis: Wie es in kompatiblen Betriebssystemen auch üblich ist, werden dem Begriff Sektor im SCPX 5105 zwei verschiedene Bedeutungen zugeordnet.

BDOS und BIOS verstehen an ihren Anwenderschnittstellen unter einem (logischen) Sektor einen 128 Bytes großen Aufzeichnungsabschnitt auf der Diskette. Diese Verfahrensweise stammt noch aus der Zeit, als es nur Disketten mit (physischen) Sektoren von 128 Bytes Länge gab.

Die (physischen) Sektoren sind dagegen die tatsächlich auf der Diskette vorhandenen Abschnitte, die mit **einem** Zugriff gelesen oder geschrieben werden müssen.

Durch die Vergrößerung der (physischen) Sektorlängen auf 256, 512, 1024 und mehr Bytes (s. auch Bedienungsanleitung ANHANG 2) müßte man eigentlich zwischen (logischen) Sektoren und (physischen) Sektoren unterscheiden. Es hat sich aber durchgesetzt, in beiden Fällen einfach vom Sektor zu sprechen.

HOME Positionieren auf Spur 0

Aufrufparameter: -

Rückkehrparameter: -

Die Routine positioniert den Kopf des selektierten Laufwerkes auf die Spur Null.

SELDSK Selektieren des Laufwerkes

Aufrufparameter: C - Laufwerksnummer

E - LOGIN-Bit (s.u.)

Rückkehrparameter: HL - Diskettenparameterkopf DPH

Die Routine wählt das im Register C angegebene Laufwerk für den nächsten Datenzugriff aus. Dabei enthält das Register C eine 0 für Laufwerk A, eine 1 für Laufwerk B usw.

Bei jedem Aufruf der Routine wird in HL die Adresse eines Speicherbereiches bereitgestellt, in dem die Merkmale der betreffenden Diskette enthalten und Platz für Zwischenergebnisse beim Diskettenzugriff vorhanden ist. Dieser Speicherbereich, als Diskettenparameterkopf bezeichnet und mit DPH (Disk Parameter Header) abgekürzt, ist für die BDOS-Arbeit vor allem deshalb wichtig, da hierüber die Informationen zur Diskettenstruktur und andere wichtige Parameter erreicht werden. Der Inhalt des DPH-Bereiches wird später beschrieben.

Wird SELDSK für ein nicht vorhandenes Laufwerk aufgerufen, dann wird HL = 0000H zurückgegeben.

Die physische Laufwerksauswahl wird erst bei einem tatsächlichen Datenzugriff (READ oder WRITE) ausgeführt.

Das niederwertigste Bit des Registers E ist 0, wenn es sich um den ersten Aufruf der Routine nach einem Kalt- oder Warmstart handelt. Dabei wird das Format der eingelegten Diskette automatisch erkannt. Diese Formaterkennung ist einer der wesentlichsten Unterschiede des SCPX 5105 zum SCP 1715!

Das selektierte Laufwerk wird erst durch erneuten Aufruf der Routine SELDSK geändert.

SETTRK Spurpositionierung

Aufrufparameter: C - Spurnummer

Rückkehrparameter: -

Die Routine bereitet die Spurpositionierung entsprechend dem Inhalt des Registers C für den nächsten Zugriff auf das ausgewählte Laufwerk vor. Im BIOS wird lediglich die

Spurnummer gemerkt und die eigentliche Positionierung bis zu einem Lese- oder Schreibbefehl zurückgestellt. Dadurch lassen sich Verwaltungsaufgaben vereinfachen. Wichtig ist lediglich, daß beim Datenzugriff der Kopf auf der richtigen Spur steht.

Die Spuradresse bleibt bis zum erneuten Aufruf der Routine SETTRK erhalten.

SETSEC Sektorpositionierung

Aufrufparameter: C - Sektornummer

Rückkehrparameter: -

Entsprechend der im Register C eingestellten Sektornummer wird auf dem ausgewählten Laufwerk der Sektor für den nächsten Diskettenzugriff positioniert. Die Sektornummer wird zunächst intern gemerkt und der eigentliche Zugriff bis zu einem Schreib- oder Lesebefehl verschoben.

Die Sektoradresse bleibt erhalten bis zu einem erneuten Aufruf der Routine SETSEC.

SCTRAN Umwandeln der Sektornummer

Aufrufparameter: BC - umzuwandelnde Sektornummer (00,...)

DE - Adresse Umwandlungstabelle

Rückkehrparameter: HL - umgewandelte Sektornummer

Die Routine erhält die logische Sektornummer in Registerpaar BC und die Adresse einer Umwandlungstabelle in Registerpaar DE.

Die logische Sektornummer (relativ zu Null angeben) wird als ein Index in der Umwandlungstabelle verwendet. Die durch die Umwandlung bestimmte physische Sektornummer wird im Registerpaar HL zurückgegeben.

Hier wird noch nichts darüber ausgesagt, ob überhaupt ein und wenn ja welcher Sektor beim nächsten Datenzugriff tatsächlich gelesen oder geschrieben wird. Es wird lediglich die Sektornummer bestimmt.

11.5.2 Datenverkehr (SETDMA, READ, WRITE)

Der Datenverkehr umfaßt neben dem Lesen und Schreiben von Informationen weiterhin die Festlegung der Adresse des Datenpuffers, jenes 128-Byte-Bereiches im Speicher, der die Daten von der Diskette übernimmt bzw. von dem sie kommen.

Liegt der Ort der Aufzeichnung auf der Diskette fest, und ist der Ort des Datenpuffers im Speicher bestimmt, dann können die Daten gelesen oder auf die Diskette geschrieben werden. Dazu bietet das BIOS die Routinen:

- SETDMA
Festlegen des Datenpuffers als Ziel oder Herkunft der Daten
- READ
Lesen eines 128-Byte-Sektors von der Diskette und übertragen in den Datenpuffer
- WRITE
Schreiben der im Datenpuffer vorliegenden Information in den adressierten 128-Byte-Sektor

SETDMA Einstellen der Pufferadresse

Aufrufparameter: BC - Sektornummer

Rückkehrparameter: -

Die Routine enthält in BC die Anfangsadresse eines 128 Bytes umfassenden Speicherbereichs, der als Datenpuffer für alle nachfolgenden Schreib- und Leseoperationen dient.

Vom Betriebssystem wird standardmäßig ein Datenpuffer der Länge 128 Bytes auf der Adresse 80H angelegt.

READ Lesen des Sektors

Aufrufparameter: -
Rückkehrparameter: A - Fehlercode

Unter der Voraussetzung, daß Laufwerk, Spur, Sektor und Adresse des Datenpuffers festgelegt wurden, versucht die Routine den durch diese Parameter bestimmten Sektor zu lesen.

Im Register A wird folgender Fehlercode zurückgegeben:

0	fehlerfreies Lesen
1	Lesen nicht möglich

Ist der Wert im Register A gleich 0, dann wird vom Betriebssystem der Diskettenzugriff als erfolgreich abgeschlossen angenommen. Tritt jedoch ein Fehler auf, dann versucht das BIOS diesen Fehler durch mehrmaliges Wiederholen zu beheben. Bei nicht behebbaren Fehlern wird vom BIOS folgende Meldung ausgegeben:

```
BIOS-Err uu: vvvvv LW=w T=xx Si=y Se=zz
```

Dabei bedeuten:

uu	- Fehlercode
vvvvv	- Zugriffsart (Read oder Write)
w	- Laufwerk (A,...)
xx	- Spur, hexadezimal (00,...)
y	- Seite (0,1)
zz	- Sektor, hexadezimal (01,...)

Wenn der Fehlercode 52 (Laufwerk nicht bereit) ausgegeben wurde, wird zusätzlich folgender Text ausgegeben:

(W-Wiederholen / sonst-Weiter)

Der Bediener hat damit zwei Möglichkeiten, den Fehler zu quittieren:

W - Wiederholen der fehlerhaften Operation
(nach Einlegen einer Diskette bzw. dem Schließen des Laufwerkshubels)

alle anderen Tasten - Ignorieren des Fehlers und Rückkehr mit Fehler in Register A

WRITE Schreiben Sektor

Aufrufparameter: C - Schreibart
(0 - Schreiben in belegten Block
1 - Schreiben in Verzeichnis
2 - Schreiben in neuen Block)

Rückkehrparameter: A - Fehlercode

Die Routine schreibt die Daten aus dem vorher festgelegten Datenpuffer auf die Diskette entsprechend den zuvor ausgewählten Parametern Laufwerk, Spur und Sektor.

Die in Register A zurückgegebenen Fehlerbedingungen und die Verfahrensweise bei Auftreten eines Fehlers sind analog der READ-Routine.

11.6. Beschreibung der Diskettenformate

Aufgrund der Vielfalt der verbreiteten Diskettenlaufwerke und Diskettenformate schließt das BIOS des SCPX 5105 die Möglichkeit der Anpassung an verschiedene Laufwerke und Diskettenformate ein. Deshalb enthält das BIOS Tabellen, die dem Nutzer (z.B. BDOS) die Disketten- und Laufwerkeigenschaften mitteilen.

11.6.1 Diskettenparameterkopf DPH

Jedem Laufwerk ist ein 16 Bytes großer Diskettenparameterkopf (DPH - Disk Parameter Header) zugeordnet, der Informationen über das Diskettenlaufwerk enthält und Arbeitsbereiche für bestimmte BDOS-Operationen einschließt.

Durch die BDOS-Routine SELDSK wird das Laufwerk ausgewählt und außerdem die Adresse des zugehörigen DPH im Registerpaar HL zurückgegeben.

Ein DPH hat folgenden Aufbau:

Byte	Name	Bedeutung
0,1	XLT	Adresse der Übersetzungstabelle für die Sektornummer. Ist die Adresse gleich 0, dann stimmen logische und physische Sektornummer überein.
2...7		Arbeitsbereiche für BDOS (reserviert).
8,9	DIRBUF	Adresse eines 128-Byte-Verzeichnispuffers. Alle DPH enthalten die gleiche Adresse.
10,11	DPB	Adresse des Diskettenparameterblockes (DPB). Jedes Laufwerk hat einen eigenen DPB.
12,13	CSV	Adresse eines Puffers, der für das Speichern eines Prüfsummenvektors zur Prüfung auf Diskettenwechsel erforderlich ist. Jedes Laufwerk hat einen eigenen Puffer.
14,15	ALV	Adresse eines Vektors, der die Diskettenbelegung widerspiegelt. Bit n gleich 1 des Vektors bedeutet, daß der Block n der Diskette von einer Datei belegt ist. Bit n gleich 0 bedeutet, daß der Block unbelegt ist. Die ersten Blöcke und damit die ersten Bits sind durch das Verzeichnis belegt. Jedes Laufwerk hat einen eigenen Vektor.

Die im DPH erfaßten Daten und Speicherbereiche werden für jedes Laufwerk getrennt bereitgestellt.

Eine Ausnahme ist der 128-Byte-Puffer für die Verzeichnisauswertung. Er kann nur einmal im System vorhanden sein, da das BDOS immer nur ein Laufwerk zur Zeit erfassen kann und bei jeder Laufwerksumschaltung das Verzeichnis neu abfragt.

11.6.2 Diskettenparameterblock DPB

Der Diskettenparameterblock (DPB) für jedes Laufwerk ist wesentlich umfangreicher. In diesem Block sind alle Informationen zusammengefaßt, die zur Verwaltung der betreffenden Diskette notwendig sind.

Dies umfaßt:

- Informationen zur Speicherkapazität und
- Informationen zur Speicherorganisation.

Der DPB enthält unter anderem:

- Angaben zur Anzahl von Sektoren pro Spur,
- Angaben zur Anzahl von Sektoren pro Block,
- Angaben zur Größe und Lage des Verzeichnisses sowie dazu, ob die Verzeichniseinträge bei jedem Zugriff auf Diskettenwechsel überprüft werden. sollen, und schließlich
- eine Angabe zur Anzahl der auf der betreffenden Diskette für das Betriebssystem reservierten (System.) Spuren.

Diese Informationen sind im DPB wie folgt festgehalten:

Byte	Name	Bedeutung																				
0,1	SPT	Sektoren pro Spur																				
2	BSH	Blockverschiebungsfaktor Darin ist die Blockgröße verschlüsselt als $\text{LOG}_2(\text{Blockgröße}/128)$ Dieser Wert stellt ein Maß für die Anzahl der Sektoren pro Block dar.																				
3	BLM	Blockmaske (widerspiegelt ebenfalls die Blockgröße). $(2^{\text{BSH}}) - 1$ Zwischen Blockgröße, Blockverschiebungsfaktor und Blockmaske bestehen folgende Beziehungen: <table border="1"> <thead> <tr> <th>Blockgröße</th> <th>BSH</th> <th>BLM</th> </tr> </thead> <tbody> <tr> <td>1024</td> <td>3</td> <td>7</td> </tr> <tr> <td>2048</td> <td>4</td> <td>15</td> </tr> <tr> <td>4096</td> <td>5</td> <td>31</td> </tr> <tr> <td>8192</td> <td>6</td> <td>63</td> </tr> <tr> <td>16384</td> <td>7</td> <td>127</td> </tr> </tbody> </table>	Blockgröße	BSH	BLM	1024	3	7	2048	4	15	4096	5	31	8192	6	63	16384	7	127		
Blockgröße	BSH	BLM																				
1024	3	7																				
2048	4	15																				
4096	5	31																				
8192	6	63																				
16384	7	127																				
4	EXM	Extentmaske Ist definiert durch die Blockgröße und die Anzahl der Blöcke pro Diskette. Ihre Größe hängt von der Organisation des Verzeichniseintrages ab. Dieser enthält als wesentlichsten Teil für die Speicherverwaltung die Nummern der jeweils belegten Blöcke: - 16 Einträge zu je 1 Byte bei weniger als 256 Blöcken pro Diskette oder - 8 Einträge zu je 2 Bytes bei mehr als 255 Blöcken pro Diskette. Im einzelnen bestehen die Beziehungen: <table border="1"> <thead> <tr> <th rowspan="2">Blockgröße</th> <th colspan="2">Extentmaske für</th> </tr> <tr> <th>mehr als 255 Blöcke</th> <th>weniger als 256 Blöcke</th> </tr> </thead> <tbody> <tr> <td>1024</td> <td>-</td> <td>0</td> </tr> <tr> <td>2048</td> <td>0</td> <td>1</td> </tr> <tr> <td>4096</td> <td>1</td> <td>3</td> </tr> <tr> <td>8192</td> <td>3</td> <td>7</td> </tr> <tr> <td>16384</td> <td>7</td> <td>15</td> </tr> </tbody> </table>	Blockgröße	Extentmaske für		mehr als 255 Blöcke	weniger als 256 Blöcke	1024	-	0	2048	0	1	4096	1	3	8192	3	7	16384	7	15
Blockgröße	Extentmaske für																					
	mehr als 255 Blöcke	weniger als 256 Blöcke																				
1024	-	0																				
2048	0	1																				
4096	1	3																				
8192	3	7																				
16384	7	15																				
5,6	DSM	Anzahl der Blöcke pro Diskette minus 1 (einschließlich des Verzeichnisses, aber ohne Systemspuren).																				
7,8	DRM	Anzahl der Verzeichniseintragungen minus 1. Die Größe einer Verzeichniseintragung beträgt 32 Bytes.																				
9,10	AL0,AL1	16-Bit-Vektor, in dem die vom Verzeichnis belegten Blöcke vermerkt sind. Dieser Vektor wird beim ersten Laufwerkszugriff an den Anfang der Belegungstabelle kopiert und dient so zur Reservierung der Verzeichnisblöcke. Er ist aus diesem Grund umgekehrt als sonst üblich organisiert: Die Zählung beginnt mit dem höchstwertigen Bit, so hat der Vektor beispielsweise bei vier Verzeichnisblöcken den Wert 0F000H.																				
11,12	CGS	Größe des Verzeichnisprüfvektors (Anzahl zu prüfender Verzeichniseintragungen dividiert durch 4)																				
13,14	OFF	Anzahl der Systemspuren Die auf OFF folgenden Bytes sind für die interne Steuerung der verschiedenen physischen Laufwerke erforderlich und dürfen nicht geändert werden!																				

ANHANG C Reservierte Speicherplätze

Die Speicherplätze zwischen 0000H und 00FFH werden als Verständigungsbereich des SCP bezeichnet und enthalten verschiedene vom Betriebssystem benutzte Daten und Anweisungen. Diese werden im folgenden zusammengefaßt dargestellt.

Speicherplätze von	bis	Inhalt
0000H	0002H	Sprung zum BIOS-Eintrittspunkt WBOOT. Damit ist ein einfacher programmierter Neustart (Sprung zu Adresse 0) möglich.
0003H		Enthält das im Abschnitt 11.4.1 beschriebene I/O-Byte.
0004H		Nummer des aktuellen Laufwerkes und Benutzernummer
0005H	0007H	Enthält eine Sprunganweisung zum BDOS. Die Sprunganweisung liefert einmal den Haupteintrittspunkt in das BDOS und zum anderen stellt die Sprungadresse die niedrigste vom Betriebssystem verwendete Speicheradresse dar. Verschiedene Hilfsprogramme verändern diese Sprungadresse, um den durch sie reduzierten Speicher zu kennzeichnen.
0008H	002FH	RST 1 bis RST 5 von SCP nicht verwendet.
0030H	0037H	RST 6 für SCP reserviert, derzeit nicht benutzt.
0038H	003AH	RST 7 enthält eine Sprunganweisung in den BIOS-Monitor (vgl. Abschnitt 6).
003BH	005BH	reserviert
005CH	007FH	durch CCP erzeugter Standard-FCB
0080H	00FFH	Standard-Datenpuffer

ANHANG D BDOS-Funktionen

Nr.	Bezeichnung	Eingang	Ausgang
0	Warmstart	---	---
1	Konsoleneingabe + Echo	---	A: Zeichen *1)
2	Konsolenausgabe	E: Zeichen (mit ^S, ^P)	---
3	Sequentielle Eingabe	---	A: Zeichen
4	Sequentielle Ausgabe	E: Zeichen	---
5	List-Ausgabe	E: Zeichen	---
6	direkte Konsolen E/A	E=FF: Status E≠FF: Ausgabe Zeichen	A: Konsolenstatus *2)
7	I/O-Byte abfragen	---	A: I/O-Byte
8	I/O-Byte belegen	E: I/O-Byte	---
9	Zeichenkette ausgeben	DE -> Kette EKZ=\$, mit ^S, ^P	
10	Eingabe Konsolpuffer	DE -> Puffer *3)	--- *4)
11	Konsolenstatus	---	A: Konsolen-Status *2)
12	Version ermitteln	---	HL: Version
13	Diskettensystem zurücksetzen	---	A: Batchmode-Flag *9)
14	Auswahl Bezugs-LW	E: LW# (0...F)	---
15	Datei eröffnen	DE -> FCB	A: DC=FF: keine Datei *7)
16	Datei schließen	DE -> FCB	A: DC=FF: keine Datei *7)
17	erste Eintragung suchen	DE -> FCB	A: DC=FF: keine Datei *7)
18	folgende Eintragung suchen	---	A: DC=FF: keine Datei *7)
19	Datei löschen	DE -> FCB	A: DC=FF: keine Datei *7)
20	nächsten Satz lesen	DE -> FCB	A: EC≠00: EOF *5)
21	nächsten Satz schreiben	DE -> FCB	A: EC≠00: Disk voll *5)
22	Datei erzeugen	DE -> FCB	A: DC=FF: Verzeichnis voll *7)
23	Datei umbenennen	DE -> FCB *8)	A: EC=FF: keine Datei *7)
24	Abfrage angeschlossener LW	---	HL:LW - Vektor
25	Abfrage Bezugs-LW	---	A: LW# (0 - 15)
26	Datenpuffer adressier.	DE -> Puffer	---
27	Belegungstabelle (ALLOC) ermitteln	---	HL -> ALLOC
28	Schutz des Bezugs-LW	---	---
29	Abfrage geschützte LW	---	HL : LW - Vektor
30	Dateimerkmale setzen	DE -> FCB	A: DC=FF: keine Datei *7)
31	Diskettenparametertabelle (DPB) ermitteln	---	HL -> DPB des selekt. LW
32	Nutzernummer abfragen/setzen	E= FF: Abfrage E≠ FF: Setzen	A: Nutzer# (0 - 15)
33	direkt adressierten Satz lesen	DE -> FCB	A: RC *6)
34	direkt adressierten Satz schreiben	DE -> FCB	A: RC *6)
35	Dateigröße berechnen	DE -> FCB	---
36	Berechnung der aktuellen Satzadresse	DE -> FCB	Eintrag in FCB+33,34,35 r0-2=fkt(ex,cr)
37	ausgewählte LW zurücksetzen	DE: LW-Vektor	A: 00
40	direkt adressierten Satz schreiben und Block initialisieren	DE -> FCB	A: RC *6)

Erläuterungen:

- * Verweis auf nachfolgende Bemerkungen
- > Die Zeichen "->„ sind zu interpretieren als "Zeiger auf"

BDOS-Interface:

Eingang: CPU-Register C : BDOS-Funktionsnummer
CPU-Register DE : Feldadressen, Vektoren
E : Eingabezeichen
UP-Ansprung : 0005H
Ausgang: CPU-Register A : Status, Zeichen
CPU-Register HL : Vektoren, Adressen

Fußnoten:

*1) Sonderzeichen

^H: Rückschritt
CR: Wagenrücklauf
LF: Zeilenschaltung
^I: Tabulation auf nächste 8. Stelle
^S: start/stop Konsolenausgabe
^P: List - Echo

*2) Status = 00 kein Zeichen im Konsoleneingabepuffer
≠ 00 Zeichen im Konsoleneingabepuffer

*3) Steuerzeichen

7F: Löschen letztes Zeichen
^C: Sprung zum Warmstart, wenn als 1. Zeichen angeboten
^E: Ende der physischen Zeile
^H: Rückschritt um 1 Zeichen
^J: (LF) Abschluß der Zeile
^M: (Abschlußtaste) Abschluß der Zeile
^X: Rücksetzen an Zeilenanfang mit Löschen

*4) Struktur des Puffers

DE: +0 | +1 | +2 | +3 | ... | +n+1
mx | nx | Z1 | Z2 | ... | Zn

mx - Kapazität des Puffers
nx - Füllstand des Puffers
Z - Zeichen im Puffer

*5) Fehler-Code (EC)

A = 00: Operation ok
≠ 00: Operation nicht ausführbar

*6) Direktzugriffs-Code (RC)

A = 0 : Lesen ungeschriebener Daten
= 03 : Extentwechsel nicht möglich
= 06 : Zugriff auf EOD-Satz der Datei

*7) Verzeichnis-Code (DC)

A = FF : Fehler
= 0,1,2,3 : Nummer des Eintrages im aktuellen Verzeichnissatz

*8) Aufbau des Dateibesreibers für Umbenennung

FCB +0 - +15 : Datei alt (LW = 0,1, ... 16)
+16 - +31 : Datei neu (LW = 0 oder LW-alt)

*9) Batchmode

A = 00 : keine Batchmodedatei
A = FF : Batchmodedatei vorhanden

Aufbau des FCB:

+00	dr	Laufwerkscodierung dr = 0 : Bezugs-LW 1 : LW A 16 : LW P
+01	f1 - f8	Dateiname in ASCII Flags in den 8. Bits f1' - f8' für Nutzerflags reserviert
+09	t1 - t3	Dateityp in ASCII Flags in den 8. Bits t1' - t3' t1' = 1/0 : geschützte/ungeschützte Datei t2' = 1/0 : SYSTEM-/Verzeichnis-Datei t3' : reserviert
+12	ex	laufende Nummer des Dateiteils (Extent) 0...31
+13	s1 - 2	reserviert für internen Gebrauch
+15	rc	Satzzähler innerhalb laufenden Extents
+16	d0 - 15	Blockverzeichnis des Extents
+32	cr	laufende Satznummer der sequentiellen Datei innerhalb des Extents
+33	r0 - 2	Satznummer der Datei für Direktzugriff
+36		Ende des FCB

Aufbau der LW-Vektoren

Vektor im CPU-Doppelregister	
bit 0	: LW A
1	: LW B
:	:
15	: LW P

ANHANG E BIOS-Funktionen

Eingang: CPU-Register C : 8-Bit-Werte
 CPU-Register BC : 16-Bit-Werte
 CPU-Register DE : zweiter 16-Bit-Wert

Ausgang: CPU-Register A : 8-Bit-Werte
 CPU-Register HL : 16-Bit-Werte

Nr.	Bezeichnung	Eingang	Ausgang
0	Kaltstartroutine	-	-
1	Warmstartroutine	-	-
2	CONSOLE-Status Eingabe abfragen	-	A: Status (FF=Zeichen da, 00=nicht da)
3	CONSOLE-Eingabe	-	A: Zeichen
4	CONSOLE-Ausgabe	C: Zeichen	-
5	LIST-Ausgabe	C: Zeichen	-
6	PUNCH-Ausgabe	C: Zeichen	-
7	READER-Eingabe	-	A: Zeichen
8	Spur Null einstellen	-	-
9	Laufwerk auswählen	C: Laufwerksnummer (0 ...) E: LOGIN-Bit (0=1.Zugriff nach ^C, 1=sonst)	HL: DPH-Adresse
10	Spur auswählen	C: Spurnummer (0,...)	-
11	Sektor auswählen	C: Sektornummer (1,...)	-
12	Datenpufferadresse setzen	BC: Pufferadresse	-
13	Selektierten Sektor lesen	-	A: Fehlercode (0=ok, 1=Fehler)
14	Selektierten Sektor schreiben	C: Schreibart (0=normal, 1=Verzeichnis, 2=neuer Block)	A: Fehlercode (0=ok, 1=Fehler)
15	LIST-Status abfragen	-	A: Status (FF=bereit, 0=nicht bereit)
16	Sektornummer umrechnen	BC: Sektornummer DE: Adr. Umwandlungstabelle	HL: Sektornummer
17	CONSOLE-Status abfragen	-	A: Status (FF=bereit, 0=nicht bereit)
18	READER-Status abfragen	-	A: Status (FF=Zeichen da, 0=nicht da)
19	PUNCHER-Status abfragen	-	A: Status (FF=bereit, 0=nicht bereit)
20	Treiber-/Slottabellenverteiler	B: Kanalnummer (0...3) C: I/O-Byte mit Subkanal	DE: Adresse Slottabellenplatz HL: Adresse Treibertabellenplatz

ANHANG F V.24-Treiber im SCPX 5105

Im Betriebssystem SCPX 5105 sind zwei verschiedene residente Treiberprogramme für V.24-Interface enthalten:

1. Druckertreiber

Hardware: PIO im CCG
Anschlußbuchse am CGG (durch DSE verdeckt),
Anschlußbuchse "PRINTER" an DSE,
Taktbereitstellung durch Software-Zeitschleife,
nur Senden möglich (Signale TxD und CTS),
nur DTR-Protokoll möglich.

Software: Treiberprogramm für Statusabfrage und Ausgabe,
Betrieb ohne Interrupt.

Standardeinstellung:

9600 Baud
8 Datenbit
kein Paritätsbit
1 Stoppbit

Diese Werte können durch ein entsprechendes Programm (*) verändert werden.

Varianten:	V24PP Ausgabe aller Zeichencodierungen (00-FF) unverändert.	V24PL Sonderbehandlung bestimmter Zeichen- codierungen, im Standardfall wird nur das Zeichen TAB (09H) in eine entsprechende Anzahl von 1 bis 8 Leerzeichen umgewandelt. Mittels Zusatzprogramm sind vielfältige Sonderfunktionen wählbar.
------------	---	--

Standardzuordnung zu den physischen Gerätenamen:

V24PP	V24PL
PUN: UP2:	LST: LPT:
LST: TTY:	

Abbruchmöglichkeit:

Das Treiberprogramm kann bei Verklemmungen (z.B. Gerät nicht angeschlossen) jederzeit durch CTRL-STOP abgebrochen werden. In der Systemzeile wird dann angezeigt:

```
V24 err: BREAK BY ^STOP
```

Danach wird ein Warmstart ausgeführt.

(*) Ein solches Installationsprogramm wird im Rahmen der Zusatzsoftware zum BIC bereitgestellt.

2. Allgemeiner V.24-Treiber

Hardware: SIO-Kanal in DSE Portadresse 69H (Daten), 6BH (Steuerung)
Anschlußbuchse "PLOTTER" an DSE.
Taktbereitstellung durch CTC Kanal 0 in DSE Portadresse 50H
Am CTC-Anschluß CLK0 liegt der durch 13 geteilte FDC-Takt (4MHz),
Senden und Empfangen möglich (Signale TxD, RxD, CTS, DTR),
Eingangssignal DCD ist fest auf logisch "1" gelegt.

Software: Treiberprogramm für Eingabe
Ausgabe
Statusabfrage Eingabe
Statusabfrage Ausgabe
DTR-Protokoll
Betrieb ohne Interrupt

Standardeinstellung:

9600 Baud
8 Datenbit
kein Paritätsbit
1 Stoppbit

Diese Werte können durch ein entsprechendes Programm verändert werden.

Standardzuordnung zu den physischen Gerätenamen:

CON: UC1: RDR: PTR: PUN: PTP: LST: UL1:

Abbruchmöglichkeit:

Das Treiberprogramm kann bei Verklemmungen (z.B. Gerät nicht angeschlossen)
jederzeit durch CTRL-STOP abgebrochen werden.
In der Systemzeile wird angezeigt:

```
U24 err: BREAK BY ^STOP
```

Danach wird ein Warmstart ausgeführt.

Fehlermeldungen

In der Systemzeile könne bei V.24-Empfang folgende Fehlermeldungen angezeigt werden:

1. Bei den folgenden drei Fehlermeldungen wird eine Tastatureingabe gefordert, die über den weiteren Programmverlauf entscheidet.

```
U24 err: PARITY (^c=abort, other=ignore)
```

Das empfangene Zeichen hat eine falsche Parität

```
U24 err: OVERFLOW (^c=abort, other=ignore)
```

Der Empfangspuffer des SIO ist übergelaufen (falsche Reaktion auf das Signal DTR).

```
U24 err: FRAME (^c=abort, other=ignore)
```

Beim Empfang eines Zeichens wurde das STOP-Bit falsch erkannt (dieser Fehler deutet auf eine falsche Einstellung der Anzahl der Daten- und Paritätsbits hin).

Bei diesen Fehlermeldungen wird nach Eingabe von ^C das gerade laufende Programm mit Warmstart abgebrochen. Bei jeder anderen Tastatureingabe wird der Fehler ignoriert und das falsch empfangene Zeichen weiterverarbeitet. Nach der Tastaturbedienung verschwindet die Fehleranzeige in der Systemzeile.

2. Bei der folgenden Fehlermeldung erfolgt automatisch ein Warmstart. Die Fehleranzeige in der Systemzeile verlischt nach einer bestimmten Zeit:

```
U24 err: PARITY (^c=abort, other=ignore)
```

Die Datenempfangsleitung RxD liegt auf "0"-Potential (Leitung unterbrochen oder Sender nicht bereit).

SACHWORTVERZEICHNIS

A

Abfrage	7.3
Abbruchanzeige	
-aktuelles Laufwerk	9.4.4
-Laufwerk im on-line-Zustand	9.4.4
-Laufwerk mit Schreibschutz	9.4.4
Adressierung	9.4.3
Allocation-Vektor	9.4.4
ALT-Taste	2.3
Anwenderprogramme im SCPX	8.1
Anzeigemodus Systemzeile	6.2
Anzeigen, hexadezimaler Inhalt	4.2.2
Anzeigen residenter Kommandos	4.1.10
Anzeigen, Textdatei	4.1.3
Attribute	
-,DIR	9.4.4
-,R/O	9.4.4
-,R/W	9.4.4
-,SYS	9.4.4
Ausgabe	
-auf Konsole	9.3.2
-auf Listgerät	9.3.3
-,gepufferte	9.3.2
-,sequentielle	9.3.1, 9.3.3
-Zeichenkette auf Konsole	9.3.2
-,zeichenorientierte	9.1, 9.3
-,zeichenweise	9.3.2
Ausgabekanäle, logische	11.4.1
AUTOEXEC.SUB	2.1, 2.2, 4.2.5, 11.3

B

Batchmode-Datei	9.4.4
Batchmode-Flag	9.4.4
BDOS	1, 8.1, 9
-,Aufgaben des	9.1
-,Fehlermeldungen des	7.2
-,Funktionen des	9, 10, Anhang D
BDOS-Ruf	9.2
BDOS-Schnittstelle	9.2
Benutzerbereich	4.1.6, 4.2, 3.8,
Benutzercode	9.4.4
Benutzerkonzept	3.8
Bereichsnummer	9.4.1
Betriebssystem	1, 2.1
Bildmodus	2.4
Bildschirm	2.4
Bildschirminhalt drucken	6.2
BIOS	1, 8.1
-,Aufgaben des	11.1
-,Fehlermeldungen des	5, 7.1
-,Funktionen des	11, Anhang D
-,Interface des	Anhang D
-,Routinen des	11.2, 11.5.2
BIOS-Mitteilungen	5
BIOS-Monitor	6.2

-verlassen	6.2
BIOS-Schnittstelle	11.2, 11.4.2
BIOS-Sprungvektor	11.2
Blockgröße	11.6.2
Blockinitialisierung	9.4.3
Blockmaske	11.6.2
BOOT	11.2, 11.3

C

CCP	1, 4, 8.1
-,Aufgaben des	4
-,Fehlermeldungen des	7.3
CONIN	11.2, 11.4.2
CONOST	11.2, 11.4.2
CONOUT	11.2, 11.4.2
CONSOLE	11.4.1
CONST	11.2, 11.4.2
CTRL-Kombinationen	2.3, Anhang A

D

Darstellungsattribute	Anhang B
Datei	3, 9.4
-,Anlegen einer	9.4.2
-,Bezeichnung einer	3.5
-,Eröffnen einer	9.4.2
-,Löschen einer	4.1.2, 9.4.2
-,Schließen einer	9.4.2
-,Umbenennen einer	4.1.5, 9.4.2
Dateiattribute	3.7, 4.2.4, 9.4.4
Dateigröße	3.4
Dateinhalt anzeigen	4.2.2
Dateikonzept	3.4
Dateiname	3.5, 9.4.1
Dateisteuerblock	9.4
Dateityp	3.5, 3.6, 9.4.1
Dateiübertragung, E/A-Kanäle	4.2.4
Dateiverwaltungsfunktionen	9.4.4
Datenpuffer	11.5.2
-,Adresse des	11.5.2
-,Laufwerk des	11.5.2
-,Sektor des	11.5.2
-,Spur des	11.5.2
Datentransport	9.4.3
Datenverkehr	11.4, 11.5.2
-über Konsole	9.3.2
-,zeichenweiser	9.3.3, 11.4
DEVTBL	11.2, 11.4.2
DIR	4.1.1
DIR-Attribut	9.4.4
Diskette	3
-,Belegung der	9.4.4, 11.6.1
-,Speicherplatzverwaltung der	3.4
Diskettenbeistellung	3.3
Diskettendatei	4.2, 9.1, 9.4
-,Kopieren von	4.2.4

- ,Verkettung von 4.2.4
- Diskettenfehler 7.1
- Diskettenformate 7.1
- Disketteninhaltsverzeichnis 4.1.1, 9.4.2
- Diskettenlaufwerk, aktuelles 3.3
- Diskettenlaufwerksauswahl 3.3
- Diskettenparameterblock 11.6.1, 11.6.2
- Diskettenparameterkopf 11.5.1, 11.6.2
- Diskettensystem rücksetzen 9.4.4
- Diskettenwechsel 4
- Diskettenzugriffsoperationen 11.1
- DMA-Adresse 8.3, 9.4.3, 9.4.4
- ,Einstellen der 9.4.3
- Drucken 4, 4.1.3
- Druckertreiber Anhang F
- DUMP 4.2.2
- ,Fehlermeldungen im 4.2.2

E

- E/A-Port
- ,Ausgabe auf 6.2
- ,Einlesen von 6.2
- Echofunktion 9.3.2
- Eingabe
- ,gepufferte 9.3.2
- in Konsolpuffer 9.3.2
- ,sequentielle 9.3.1, 9.3.3
- von Konsole 9.3.2
- ,zeichenorientierte 9.1, 9.3
- ,zeichenweise 9.3.2
- Eingabekanäle, logische 11.4.1
- Eingabeparameter 9.2
- Eingabezeile 4
- Eintragung 9.4.2
- ,Suchen nach erster 9.4.2
- ,Suchen nach nächster 9.4.2
- ERA 4.1.2
- ESC-Folgen Anhang A, Anhang B
- EXT 4.1.7
- Extent 9.4.1, 9.4.3
- Extentgrenze 9.4.3
- Extentmaske 11.6.2
- Extentnummer 9.4.1, 9.4.3

F

- FCB 8.3, 9.4.2
- ,Aufbau des 9.4.1, Anhang D
- Fehlercode 11.5.2
- Fehlermeldungen
- ,BDOS 7.2
- ,BIOS 5, 7.1
- ,CCP 7.3
- ,DIR 4.1.1
- ,Diskette 7.1
- ,DUMP 4.2.2
- ,ERA 4.1.2
- ,EXT 4.1.7

- ,Kommandoeingabe 7.3
- ,Laden transienter Kommandos 7.3
- ,PIP 4.2.4
- ,REN 4.1.5
- ,STAT 4.2.3
- ,SUBM 4.2.5
- ,V.24-Empfang Anhang F
- ,zeichenweiser Datenverkehr 7.1
- ,ZG 4.2.7
- FORMAT 4.2.1
- Formaterkennung 11.5.1
- Formatieren 3.1, 4.2.1
- Fragezeichen ? 3.5
- Funktionstasten 2.3, Anhang A
- ,Anzeige der 2.3, 5
- ,Belegung der 2.3, Anhang A
- ,Einstellen der 6.2
- ,programmierbare 2.3, Anhang A
- Füllstand 9.4.1

G

- Geräte
- ,CONSOLE 11.1
- ,im PIP 4.2.4
- ,LIST 11.1
- ,logische 4.2, 7.1, 8.2, 11.4
- ,physische 4.2
- ,PUNCH 11.1
- ,READER 11.1
- Gerätetreiber, physischer 11.4.1
- GO 4.1.11

H

- HELP 4.1.10
- Hilfsfunktionen 9.4.4
- HOME 11.2, 11.5.1

I

- Inhaltsverzeichnis 2.2, 3.2, 4.2.5, 9.4.3
- INIT 4.2.1
- Initialisierung 9.4.4, 11.3
- Interrupt 8.1
- I/O-Byte 4.2.3, 9.3.1
- ,Abfragen des 9.3.1
- ,Aufbau des 11.4.1
- ,Setzen des 9.3.1

K

- Kaltstart 2.2, 3.2, 4.2.5, 9.4.3
- Kaltstartkommando 11.3
- Kaltstartroutine 11.1, 11.3
- Kanäle 9.3
- Kanalbezeichnungen

- ,CON	11.4.1
- ,LST	11.4.1
- ,PUN	11.4.1
- ,RDR	11.4.1
Kommando(s)	
- ,Darstellung der	4
- ,DUMP	4.2.2
- ,PIP	4.2.4
- ,residente	4, 4.1
- ,residente anzeigen	4.1.10
- ,resident machen	4.1.7
- ,STAT	4.2.3
- ,SUBM	4.2.5
- ,transiente	4, 4.2
- ,XSUBBIC	4.2.6
- ,ZG	4.2.7
Kommandoeingabe (CCP)	8.3
- ,Fehlermeldungen	7.3
Konsole(n)	9.3
- ,Ausgabe auf	9.3.2
- ,Ausgabe, direkt	9.3.2
- ,Eingabe von	9.3.2
- ,Eingabe, direkt	9.3.2
Kopieren	4.2.4
-Dateiausschnitte	4.2.4
-Diskettendatei	4.2.4

L

Laufwerk	9.4.4
- ,aktuelles	2.1, 2.2, 3.3, 9.4.4
- ,ausgezeichnetes	9.4.4
- ,logisches	3.3, 9.4.4
- ,Selektieren eines	9.4.4, 11.5.1
- ,Vertauschen von	4.1.9
- ,Zurücksetzen von	9.4.4
Laufwerksfunktionen	9.4.4
Laufwerksparameter	9.4.4
Laufwerksauswahl	3.3
Laufwerksbezeichnung	3.3
Laufwerkscode	9.4.1
Laufwerkssteuerung	11.5.1
Laufwerkswechsel	4
Laufwerksvektoren	Anhang D
Leistungsfähigkeit SCPX	9.4.4
Lesen	
- ,mit direktem Zugriff	9.4.3
- ,sequentielles	9.4.3
LIST	1.2, 11.4.1, 11.4.2
Listendruck (PIP)	4.2.4
Listgerät	9.3.1, 9.3.3
LISTST	11.2, 11.4.2
LOGIN	11.5.1

O

OFF-LINE-Zustand	9.4.4
ON-LINE-Zustand	9.4.4

P

PIP	4.2.4
- ,Fehlermeldungen im	4.2.4
- ,Parameter des	4.2.4
Positionierung	11.5.1
- ,Sektor	11.5.1
- ,Spur	11.5.1
Prüfsummenvektor	11.6.1
Prüfvektor	
- ,Verzeichnis des	11.6.2
Pseudografikzeichen	Anhang B
Pufferadresse einstellen	11.5.2
PUNCH	11.2, 11.4.1, 11.4.2
PUNST	11.3, 11.4.2

R

READ	11.2, 11.5.2
READER	11.2, 11.4.1, 11.4.2
READST	11.2, 11.4.2
Registerinhalt	
- ,Anzeigen	6.2
- ,Ändern	6.2
REN	4.1.5
RES	4.1.8
Reservierte Speicherplätze	Anhang C
R/O-Attribut	9.4.4
R/W-Attribut	9.4.4

S

Satz	9.4.2
Satzadresse	9.4.3
Satznummer	9.4.1, 9.4.3
Satzposition, aktuelle	9.4.3
Satzposition, freie	9.4.4
Satzzeiger	9.4.3
SAVE	4.1.4
Schreiben	
-mit direktem Zugriff	9.4.3
-mit direktem Zugriff und	
Blockinitialisierung	9.4.3
- ,sequentielle	9.4.3
Schreibschutz	9.4.4
SECTRAN	11.2, 11.5.1
Sektor(en)	11.5.1
-adressieren	11.5.1
-lesen	11.5.2
- ,logischer	11.5.1
- ,physischer	11.5.1
-pro Block	11.6.2
-pro Spur	11.6.2
-schreiben	11.5.2
Sektornummer umwandeln	11.5.1
Sektorpositionierung	11.5.1
SELDSK	11.2, 11.5.1
SETDMA	11.2, 11.5.2
SETSEC	11.2, 11.5.1

SETTRK	11.2, 11.5.1	Verzeichniscode	9.4.4
SGENBIC.BAS	3.2	Verzeichniseintragung	9.4.2
Sonderzeichen	Anhang B	Verzeichnispuffer	11.6.1
Speicherinhalt		Verzeichnis-Prüfvektor	11.6.2
-,Anzeigen des	6.2	V.24-Treiber	Anhang F
-,Ändern des	6.2	-,allgemeiner	Anhang F
Speicherkapazität	11.6.2	-,Drucker	Anhang F
Speicherkonfiguration	8.1		
Speichern	4.1.4	W	
Speicherorganisation	11.6.2	Warmstart	2.2, 9.4.3, 9.4.4
Speicherverwaltungsschaltkreis (SVS)	8.1	Warmstartroutine	11.1, 11.3
Sprungvektor	11.1	Warteschleife	6.1
Spurpositionierung	11.5.1	WBOOT	11.2, 11.3
Stack	8.3	WRITE	11.2, 11.5.2
STAT	4.2.3		
-,Fehlermeldungen im	4.2.3	X	
Stern *	3.5	XSUBBIC	4.2.6
Steuerzeichen	Anhang B		
STOP-Taste	6	Z	
STOP-Zustand	6	Zeichenausgabeoperation	11.1
Subkanal	4.2.3, 9.3.1, 11.4.1	Zeichendarstellung	2.4
SUBM	2.1, 4.2.5, 11.3	Zeicheneingabeoperation	11.1
-,Fehlermeldungen im	4.2.5	Zeichengenerator	4.2.7
SWAP	4.1.9	Zeichenkette	9.3.2
Syntaxerläuterung	4	Zeichensatz	2.3, 2.4, Anhang B
SYS-Attribut	9.4.4	Zeilen-Editor-Funktion	4
Systemdatei	2.1, 2.2, 3.2, 11.3	ZG	4.2.7
Systemdiskette	3.2, 4.2	-,Fehlermeldungen im	4.2.7
Systemfunktionen	9.4.4	Zugriff	
Systeminitialisierung	11.1, 11.3	-,direkter	9.4.3
Systemschnittstelle	8.1	-,sequentieller	9.4.3
Systemspuren	11.6.2		
Systemstatus	5		
Systemverwaltung	9.1		
System-Warmstart	4.2.3		

T

Tastatur	2.3
Tastencode	Anhang A
Textdatei anzeigen	4.1.3
TPA	8.1
Treiberinstallation	11.4.2
TYPE	4.1.3

U

Unterbrechung	6.1
USER	3.8, 4.1.6, 4.2, 9.4.4
USER-Konzept	3.8
Übersetzungstabelle	11.6.1

V

Verketten, Diskettendateien	4.2.4
Versionsnummer	9.4.4
Verständigungsbereich	8.1, 9.2, Anhang C
Verzeichnis	9.4.2