

robotron

Betriebssystem

KC85/1

(Z9001)

Herausgeber:
VEB Robotron-Meßelektronik
„Otto Schön“ Dresden

Inhaltsverzeichnis

1.	Allgemeiner Überblick	1
1.1.	Bestandteile des Monitors	1
1.2.	Speicheraufteilung	3
2.	Monitormodule	7
2.1.	CONSOLE COMMAND PROGRAMM (CCP)	7
2.2.	BASIC OPERATING SYSTEM (BOS)	9
2.3.	BASIC INPUT/OUTPUT SYSTEM (BIOS)	23
3.	Zusätzliche Treiber	28
3.1.	Treiberinitialisierung	28
3.2.	Besonderheiten der Treiber für CONST	29
3.3.	Besonderheiten der Treiber für READER und LIST	31
4.	Spezielle Monitorroutinen	32
4.1.	Monitorroutinen im CCP	32
4.2.	Monitorroutinen im BOS/BIOS	36
4.3.	Monitorroutine zur Kassetten-E/A	42

Anlage: Quelltext

1. Allgemeiner Überblick

Vorbild für den Monitor des Computers Z 9001 ist das Betriebssystem CP/M-80. Allgemeiner Aufbau und externe Schnittstellen wurden weitestgehend vom Vorbild übernommen. Entscheidende Abweichungen gibt es bei den residenten Kommandos und der Arbeit mit externen Datenträgern. Die im CP/M-80 üblichen residenten Kommandos wurden durch Z 9001-spezifische ersetzt, und die Dateiarbeit wurde den Möglichkeiten und Anforderungserfordernissen des Computers angepaßt (nur Kassette als externer Datenträger). Dementsprechend wurden Systemrufe und Direktrufe des CP/M-80 zur Diskettensteuerung durch spezielle, auf den Z 9001 zugeschnittene, ersetzt.

Um den Monitor des Computers Z 9001 in einer späteren Ausbaustufe entscheidend aufwerten zu können, besteht die Möglichkeit, den Monitor um beliebige Kommandos und Gerätetreiber zu erweitern. Die entsprechenden Programme sind sowohl in RAM- als auch in ROM-Versionen arbeitsfähig. Zur Systemerweiterung bzw. Modifizierung ist der Monitor in der Lage, nach Einschalt-RESET/RESET ein Programm mit dem Namen "#" im Speicher (RAM/ROM) zu suchen und automatisch zu starten (siehe 2.1.2. und 3.).

1.1. Bestandteile des Monitors

Entsprechend dem Vorbildsystem CP/M-80 wurde auch der Monitor des Computers Z 9001 in drei Hauptbestandteilen realisiert. Diese Bestandteile sind jedoch im Gegensatz zum Vorbildsystem lediglich funktionell eindeutig voneinander zu trennen, da der für den Monitor zur Verfügung stehende Speicherbereich sehr begrenzt ist. Der Monitor des Computers Z 9001 besitzt folgende Hauptbestandteile:

a) CONSOLE COMMAND PROGRAMM (CCP)

Funktion:

- Steuerung des Computers im OS-Mode
- Kommando- und Parameterübernahme
- Verzweigen zur Kommandoausführung
- Ausführen der residenten Kommandos
- Verwaltung der Gerätetreiber

Inhalt:

- Monitorgrundschleife zur Kommandoanforderung vom Nutzer
- Kommandoentschlüsselung
- Routinen zur Ausführung der residenten Kommandos
- Subroutinen zur Kommando- und Parameterübernahme und deren Prüfung
- Subroutinen für Speichertest und Speicherverwaltung
- Subroutinen zum Test auf zulässige Geräteadressierung bei Zeichen-E/A
- allgemeine Hilfsroutinen

Eintritt nach:

- Einschalten des Computers
- RESET
- Systemruf 0 und JMP 0
- Betätigung der STOP-Taste im OS-Mode
- Rückkehr von der Kommandoausführung
- Rückkehr aus einem Anwenderprogramm

b) BASIC OPERATING SYSTEM (BOS)

Funktion:

- Steuerprogramm für die Ausführung von Systemrufen
- Prüfen der Systemrufe
- Übernahme von Parametern
- Verzweigen zur Ausführung
- Rückgabe von Parametern
- Ausgabe von Fehlermeldungen

Inhalt:

- zentraler Eintrittspunkt zur Rufnummernentschlüsselung, Parameterübernahme und Registerrettung
- Adreßvektor zur Adressierung der angewählten Systemprogramme
- Systemprogramme zur Zeichenketten- und Kassetten-E/A -Fehlerbehandlung
- zentraler Austrittspunkt mit Rückgabe von Parametern, Ausgabe von Fehlermeldungen und Registerregenerierung

Eintritt:

- nach CALL 0005
- Rufnummer in Register C $C = 0(1) \dots 33$
- Parameter in Register DE bzw. A

Austritt:

- Parameter in Reg BC bzw. A

c) BASIC INPUT/OUTPUT SYSTEM (BIOS)

Funktion:

- übersetzt Rufe von CCP und BOS durch Unterprogramme zur Einzelzeichen-E/A
- verbindet logische und physische Geräte und übergibt die Zeichen den entsprechenden Treiberprogrammen
- übergibt übergeordneten Steuerprogrammen Parameter und spezielle Fehlercodes

Inhalt:

- Sprungtabelle zur Direktadressierung spezieller Monitorprogramme
- Programme zur Initialisierung des Computers nach Kalt- bzw. Warmstart
- Programme zur Einzelzeichen-E/A
- Standardtreiber für Tastatur und Bildschirm
- Standardtreiber für Kassetten-E/A
- spezielle Unterprogramme zur Ausführung von System- und Direktrufen (werden von BOS zum Teil unmittelbar mitgenutzt)
- verschiedene Hilfsprogramme (zum Teil von CCP und BOS mitgenutzt)

Eintritt:

- nach CALL 0F000H + (n*3) $n = 0(1) \dots 22$
- Parameter in BC bzw. C

Austritt

- Parameter in BC bzw. A
- Fehlerstatus in CY

1.2. Speicheraufteilung

Die Speicheraufteilung des Computers Z 9001 weicht erheblich von der des CP/M-80 ab. Alle Systemzellen wurden in den Bereich bis 0FFH verlegt. Die erste durch den Anwender nutzbare Adresse ist 300H. Interne Puffer und der Stackbereich liegen im Bereich von 100H bis 2FFH.

MONITOR
BWS

ROM Erweiterung (z.B. BASIC)
RAM 2.Erweiterung
RAM 1.Erweiterung
RAM Grundgerät
System -RAM

- Speicher Gesamtüberblick -

Bei der Aufteilung des Speichers wurde berücksichtigt, daß auf den RESTART-Adressen jeweils 3 Bytes für die Aufnahme eines Sprungbefehls frei bleiben, um die Arbeit der Anwender mit den RST-Befehlen zu ermöglichen. Eine Ausnahme bilden die Adressen 0 und 66H. Die Adressen 0 bis 2 sind mit einem Sprung zum Warmstart belegt, so daß der Befehl RST 0 die Übergabe der Steuerung an das CCP zur Folge hat. Die NMI-Adresse 66H ist für den Z9001 ohne Bedeutung, da für die Behandlung eines NMI keine Notwendigkeit bestand. Die Adresse 66H wird daher während der Kassetten-E/A mitgenutzt. Zusätzlich wurden im nicht angezeigten Teil des BildwiederholSpeichers einige Systemzellen für den Systemschutz und die aktuelle Gerätekonfiguration eingerichtet (EFC0H).

1.2.1. Systemspeicher

Adresse	Befehl		
0000	JMP	WBOOT	;Sprung zum Warmstart
0003	BER	1	;frei
0004	IOBYT:	DB 1	;I/O-Byte zur Zuweisung log.Gerät phy. Gerät
0005	JMP	BOS	;Sprung zum BOS-Eintritt (für alle Systemrufe)
0008	BER	3	;frei für RST 8
000B	SPSV:	DA 0	;Register für aktuellen Steck bei Eintritt in BOS
000D	BCSV:	DA 0	;Register für BC bei Eintritt in BOS
000F	ASV:	DB 0	;Register für A bei Eintritt in BOS
0010	BER	3	;frei für RST 10H
0013	JOYR:	DB 0	;Datenpuffer für Spielbebel 1
0014	JOY1:	DB 0	;Datenpuffer für Spielhebel 2
0015	LISW:	DB 0	;Schalter für Hardcopy ;0 kein Copy ;1 Copy
0016	BSW:	DB 0	;Schalter für Kontrollton ;0 kein Kontrollton ;1 Kontrollton
0017	COLSW:	DB 0	;Merkzelle für Farbsteuercode
0018	BER	3	;frei für RST 18H
001B	DMA:	DA 80H	;Zeiger auf Puffer für Kassetten-E/A
001D	HOUR:	BER 1	;Puffer für Stunden
001E	MIN:	BER 1	;Puffer für Minuten

Adresse	Befehl		
001F	SEC:	BER 1	;Puffer für Sekunden
0020		BER 3	;frei für RST 20H
0023	COUNT:	DB 0	;Zähler zur Tastaturentprellung und ;REPEAT-Funktion der Tastatur
0024	LAKEY:DB	0	;Merkzelle für letztes gültiges Zeichen von Tastatur
0025	KEYBU:	DB 0	;Tastaturpuffer
0026	SHLOC:	DB 0	;Schalter für SHIFT LOCK ;0 kein SHIFT LOCK ;1 SHIFT LOCK
0027	ATRI:	DB 2	;aktuelles Farbattribut
0018		BER 3	;frei für RST 28H
002B	CHARP:	DB 1	;aktuelle Spalte d. Cursors(1-40)
002C	LINEP:	DB 1	;aktuelle Zeile d. Cursors(1-24)
002D	CURS:	DA 0EC00H	;aktuelle physische Adresse des Cursors
002F	PU:	BER 1	;Hilfszelle
0030		BER 3	;frei für RST 30H
0033	WORKA:	BER 1	;Hilfszelle
0034	BUFFA:	BER 1	;Puffer für das Attribut des von Cursor überlagerten ;Zeichens
0035	BU:	BER 1	;Hilfszelle
0036	EOR:	BER 2	;Zeiger auf letzte für Anwender freie Adresse
0038		BER 3	;frei für RST 38H
003B	P1ROL:	DB 0	;1. rollende Zeile - 1
003C	P2ROL:	DB 25	;letzte zu rollende Zeile + 1
003D	P3ROL:	DB 0	;1. zu rollende Spalte - 1
003E	P4ROL:	DB 41	;letzte zu rollende Spalte + 1
003F	BUFF:	BER 1	;Puffer für das vom Cursor überschriebene Zeichen
0040	PARBU:	BER 2	;Hilfszelle zur Paramterpufferung
0042		BER 26	;frei für IDAS, DEBUGGER, usw.
005C	FCB:	BER 36	;Dateikontrollblock
	FNAME:	EQU FCB	;Dateiname 8 Zeichen
	FTYP:	EQU FCB+8	;Dateityp 3 Zeichen
	PSUM:	EQU FCB+13	;Prüfsumme eines Datenblockes
	ARB:	EQU FCB+14	;Hilfszelle für Kassettentreiber
	BLNR:	EQU FCB+15	;Blocknummer
	LBLNR:	EQU FCB+16	;gesuchte Blocknummer bei Lesen
	AADR	EQU FCB+17	;Dateianfangsadresse
	EADR	EQU FCB+19	;Dateiendeadresse
	SADR	EQU FCB+21	;Startadresse, wenn Datei ein Maschinencode- ;programm ist
	SBY:	EQU FCB+23	;Schutzbyte ;0 nicht geschützt ;1 System nach Laden der Datei vor WRITE ; geschützt
0080	CONBU:	DB 80	;CCP-Eingabepuffer und Standardpuffer für ;Kassetten-E/A
0081		DB 0	;aktuelle Länge der Eingabezeile
0082		BER 80	;
00D2		BER 46	;frei
0100	INTLN:	DB 0	;interner Zeichenkettenpuffer
0101		BER 80	
0151		BER 6FH	
01C0	BOSSP:	BER 40H	;Beginn BOS-Stackbereich
	CCPSP:	EQU #	;Beginn CCP- und Nutzerstackbereich
	INTV:	EQU #	;Interruptadresstabelle
0200		DA IKACT	;Kassette schreiben
0202		DA 0	;frei für Anwender CTC Kanal 1

Adresse	Befehl		
0204	DA	ICTC	;Entprellen Tastatur, Vorteiler für Systemuhr
0206	DA	INUHR	;Sekundentakt Systemuhr
0208	DA	INTP	;Tastaturinterrupt
020A	DA	IKEP	;Kassette lesen
020C	BER	74H	;frei für Anwenderinterrupts
;der folgende Speicherbereich ist durch den Nutzer für eigene Treiber verwendbar			
0280	RTTYC	BER 4	;RAM TTY-Treiber für CONST
0284	RCRTC:	BER 4	;RAM CRT-Treiber für CONST
0288	RBATC:	BER 4	;RAM BAT-Treiber für CONST
028C	RUC:	BER 4	;RAM UC-Treiber für CONST
0290	PTTYR:	BER 4	;RAM TTY-Treiber für READER
0294	RRDR:	BER 4	;RAM RDR-Treiber für READER
0298	RUR1:	BER 4	;RAM UR1-Treiber für READER
029C	RUR2:	BER 4	;RAM UR2-Treiber für READER
02A0	RTTYP:	BER 4	;RAM TTY-Treiber für PUNCH
02A4	RPUN:	BER 4	;RAM PUN-Treiber für PUNCH
02A8	RUP1:	BER 4	;RAM UP1-Treiber für PUNCH
02AC	RUP2:	BER 4	;RAM UP2-Treiber für PUNCH
02B0	RTTYL:	BER 4	;RAM TTY-Treiber für LIST
02B4	RCRTL:	BER 4	;RAM CRT-Treiber für LIST
02B8	RLST:	BER 4	;RAM LST-Treiber für LIST
02BC	RUL:	BER 4	;RAM UL-Treiber für LIST
02C0	SYSR:	BER 40H	;reserviert für Systemerweiterung

1.2.2. Bildwiederholtspeicher

Der gesamte Bildwiederholtspeicher besteht aus zwei Teilen.

- a) Zeichenspeicher
 - 1k Bytes RAM ab Adresse EC00H
 - es werden 960 Bytes zur Anzeige gebracht
- b) Farbspeicher (nur mit Farbzusatzmodul)
 - 1k Bytes RAM ab Adresse EB00H
 - es werden 960 Bytes zur Anzeige gebracht

Die Adresse des Farbattributes im Farbspeicher ergibt sich aus der Adresse des zugehörigen Zeichens im Zeichenspeicher minus 400H.

Aufbau des Farbattributes:

Bit	Funktion	

0	rot	Zeichenhintergrund
1	grün	Zeichenhintergrund
2	blau	Zeichenhintergrund
3		nicht genutzt
4	rot	Zeichenvordergrund
5	grün	Zeichenvordergrund
6	blau	Zeichenvordergrund
7		Zeichenblinken

Hinweis:

Bit 3 im Farbattribut eines Zeichens wird in keiner Weise genutzt und kann durch den Anwender belegt werden. Die im Zeichenspeicher freien 64 Bytes werden durch den Monitor als Systemspeicher mitgenutzt. Sie dienen der Speicher- und Treiberverwaltung.

Speicherbelegung im Zeichenspeicher:

Adresse	Befehl		
EFC0	SYSB:	DB	0 ;Systembyte
EFC1	MAPAR:	BER	8 ;64-Bit-Register für Speicherkonfiguration ;ein Bit je 1k Speicher ;0 ROM bzw. geschützt ;1 RAM
;Adreßtabelle der Gerätetreiber			
EFC9	ATTYC:	DA	-1 ;Adresse TTY-Treiber für CONST
EFCB	ACRTC:	DA	CRT ;Adresse CRT-Treiber für CONST
EFCD	ABATC:	DA	BAT ;Adresse BAT-Treiber für CONST
EFCF	AUC:	DA	-1 ;Adresse UC-Treiber für CONST
EFD1	ATTYR:	DA	-1 ;Adresse TTY-Treiber für READER
EFD3	ARDR:	DA	-1 ;Adresse RDR-Treiber für READER
EFD5	AUR1:	DA	-1 ;Adresse UR1-Treiber für READER
EFD7	AUR2:	DA	-1 ;Adresse UR2-Treiber für READER
EFD9	ATTYP:	DA	-1 ;Adresse TTY-Treiber für PUNCH
EFDB	APUN:	DA	-1 ;Adresse PUN-Treiber für PUNCH
EFDD	AUP1:	DA	-1 ;Adresse UP1-Treiber für PUNCH
EFDF	AUP2:	DA	-1 ;Adresse UP2-Treiber für PUNCH
EFE1	ATTYL:	DA	-1 ;Adresse TTY-Treiber für LIST
EFE3	ACRTL:	DA	-1 ;Adresse CRT-Treiber für LIST
EFE5	ALST:	DA	-1 ;Adresse LST-Treiber für LIST
EFE7	AUL:	DA	-1 ;Adresse UL-Treiber für LIST
;Zeichenkettenadreßtabelle			
EFEP	TXCON:	DA	PHYDV+2 ;Adresse einer Zeichenkette des aktuellen CONST- ;Gerätes, Zeichenkette wird im ASGN-Kommando ;ausgegeben
EFEB	TXRDR:	DA	PHYDV ;
EFED	TXPUN:	DA	PHYDV ;entsprechend TXCON
EFEF	TXLST:	DA	PHYDV ;

2. Monitormodule

2.1. CONSOLE COMMAND PROGRAMM (CCP)

Das CCP ist das zentrale Steuerprogramm des Computers Z 9001. Hier erfolgt die Anforderung, Übernahme und Prüfung aller Kommandos. Residente Kommandos werden durch das CCP selbst ausgeführt. Bei transienten Kommandos wird zu deren Ausführung verzweigt. Wird ein Kommando eingegeben, welches weder ein residentes noch ein transientes Kommando ist, so wird das eingegebene Kommando als Name (Aufbau siehe CLOAD-Kommando) eines Anwenderprogrammes interpretiert. Das Programm wird von der Kassette eingelesen und entsprechend den Informationen im aktualisierten FCB plazierte und gestartet.

Bei Rückkehr mit Fehler in die Monitorgrundschleife wird entsprechend dem Fehlercode eine Fehlerausschrift ausgegeben (siehe 2.2.3.). Das CCP kann durch den Anwender umgangen werden, indem im Speicher das transiente Kommando "#" abgelegt wird (siehe 2.1.2.). In diesem Fall verzweigt die Initialisierungsroutine WBOOT automatisch zur Ausführung dieses Kommandos, unter Umgehung des CCPs. Damit ist es möglich, den Monitor abzuschalten und die Steuerung durch ein eigenes Programm selbst zu übernehmen.

Das CCP besitzt einen eigenen Stackbereich ab 200H in einer Länge von 40H. Dies ist gleichzeitig der Stackbereich eines Anwenderprogrammes, sofern dieses nicht einen eigenen Stack einrichtet. Die für den Anwender verfügbare Stacktiefe ist dabei 3CH.

Mit Eintritt in ein Anwenderprogramm ist SP = 1FCH. Auf Adresse 1FCH steht die Adresse der Fehleranzeige (ERDIS), auf Adresse 1FEH die Adresse des CCPs (GOCPM).

2.1.1. Residente Kommandos

T I M E

Funktion 1:

- Anzeige der aktuellen Systemzeit

Syntax

TIME (ENTER)

Funktion 2:

- Stellen der Systemuhr

Syntax

TIME Stunde [, Minute [, Sekunde]] (Enter)

- Eingabeparameter dezimal ohne Suffix
- für weggelassene Parameter wird 00 ergänzt

C L O A D

Funktion:

- Einlesen einer Datei (Daten, Programme)
- Speichern ab der im FCB gelesenen Anfangsadresse
- kein Start von Programmen

Syntax

CLOAD Name [.Typ] (Enter)

- Name maximal 8 Zeichen mit Buchstaben beginnend
- Typ maximal 3 Zeichen mit Buchstaben beginnend
- bei Weglassen des Typs wird eine Datei vom Typ .com erwartet

A S G N

Funktion 1:

- Anzeige der aktuellen Gerätezuweisung

Syntax

ASGN (Enter)

Funktion 2:

- Zuweisung von Standardtastatur und -bildschirm zu CONST

Syntax

ASGN CONST:=CRT (Enter)

Funktion 3:

- CONST-Gerät in BATCH-Modus schalten
- Konsoleingabe erfolgt über das aktuelle READER-Gerät
- Konsolenausgabe erfolgt über das aktuelle LIST-Gerät
- READER und LIST müssen zuvor zugewiesen werden

Syntax

ASGN CONST:=BAT (Enter)

Funktion 4:

- Zuweisen des Standardbildschirmes zum LIST-Gerät

Syntax

ASGN LIST:=CRT (Enter)

Funktion 5:

- Zuweisung von Anwendertreibern
- Treiber kann als Programm auf Kassette oder als transientes Kommando ausgelegt sein

Syntax

ASGN log.Gerät:=Name (Enter)

- logische Geräte sind CONST, READER, PUNCH und LIST

2.1.2. Transiente Kommandos

Der Monitor des Z 9001 erlaubt es dem Anwender, den Kommandovorrat um beliebige Nutzerkommandos zu erweitern. Diese Kommandos können sowohl im RAM, als auch im ROM platziert werden. Die Programmierung dieser Kommandos unterliegt einem speziellen Schema. Das

Kommando muß auf einer integralen 100H-Grenze beginnen. Beliebige viele Kommandos können in einer Tabelle angegeben werden.

Beispiel 1:

```
;Einzelkommando
    JMP    AUSF          ;Sprung zur Kommandoausführung
    DB     'NAME '       ;Kommandoname (im OS-Mode einzugeben)
                                ;8 Zeichen
                                ;ggf. mit Leerzeichen auffüllen
    DA     0              ;Endekennzeichen
```

Beispiel 2:

```
;Kommandotabelle
    JMP    AUSF1          ;Sprung zur Ausführung Kommando 1
    DB     'Name1 '       ;Kommandoname 1
    DB     0              ;Ende Kommando 1
    JMP    AUSF2          ;Sprung zur Ausführung Kommando 2
    DB     'Name2 '       ;Kommandoname 2
    DB     0              ;Ende Kommando 2
    .....
    JMP    AUSFN          ;Sprung zur Ausführung Kommando N
    DB     'NameN '       ;Kommandoname N
    DA     0              ;Ende Kommandotabelle
```

Hinweis:

- derartige Kommandos können auch bei der Zuweisung von Geräten verwendet werden
- dabei ist, an Stelle des Sprunges zur Kommandoausführung, ein Sprung zur Treiberinitialisierung zu programmieren

Eine Besonderheit bildet der Kommandoname '# '. Dieses Kommando dient zur Systemerweiterung. In der Initialisierungsroutine des Computers wird der Speicher nach diesem Kommando durchsucht. Wird dabei dieses Kommando gefunden, so springt die Initialisierungsroutine unter Umgehung des Monitors direkt zu dessen Ausführung.

2.2. BASIC OPERATING SYSTEM (BOS)

2.2.1. Überblick über die Systemrufe

Das BOS ist ein Steuerprogramm zur Ausführung von speziellen Systemrufen des Anwenders. Mit diesen Systemrufen ist es möglich, relativ komplexe Funktionen durch den Monitor ausführen zu lassen, ohne dessen interne Schnittstellen und Funktionen zu kennen. Der Aufruf von BOS erfolgt über CALL 0005. Die Auswahl des gewünschten Systemrufes erfolgt über das C-Register, dessen Inhalt den Systemruf adressiert. Verschiedene Systemrufe erwarten Eingabeparameter bzw. liefern Parameter zurück.

Eingabeparameter:

- Bytewerte im E -Register
- Wortwerte im DE-Register

Ausgabeparameter:

- Bytewerte im A -Register
- Wortwerte im BC-Register

BOS kontrolliert selbständig die korrekte Ausführung des Systemrufes. Im Fehlerfalle wird eine geeignete Fehlermeldung ausgegeben, die Rückkehr erfolgt mit CY = 1 und einem entsprechenden Fehlercode im A-Register (siehe 2.2.3.).

Alle Register außer Index- und Hintergrundregister werden gerettet, soweit sie nicht Parameter oder einen Fehlercode zurückvermitteln. Außerdem besitzt das BOS einen eigenen Stackbereich ab 1C0H.

Tabelle der Systemrufe:

Rufnr.	Name	Funktion
00	INIT	Ausführung eines Kaltstartes
01	CONSI	Eingabe eines Zeichens von CONST
02	CONSO	Ausgabe eines Zeichens zu CONST
03	READI	Eingabe eines Zeichens von READ
04	PUNO	Ausgabe eines Zeichens zu PUNCH
05	LISTO	Ausgabe eines Zeichens zu LIST
06	GETST	Abfrage der Spielbebel
07	GETIO	Lesen I/O-Byte
08	SETIO	Setzen I/O-Byte
09	PRNST	Ausgabe einer Zeichenkette zu CONST
10	RCONB	Eingabe einer Zeichenkette von CONST
11	CSTS	Abfrage Status CONST
12	RETVN	Abfrage der Versionsnummer des Monitors
13	OPENR	Eröffnen Kassette lesen
14	CLOSR	Abschließen Kassette lesen
15	OPENW	Eröffnen Kassette schreiben
16	CLOSW	Abschließen Kassette schreiben
17	GETCU	Abfrage logische und pyhsische Cursoradresse
18	SETCU	Setzen logische Cursoradresse
19	BOSER	nicht benutzt
20	READS	Lesen eines Blockes von Kassette
21	WRITS	Schreiben eines Blockes auf Kassette
22	SETTI	Stellen der Systemuhr
23	GETTI	Abfrage der Systemuhr
24	PRITI	Ausgabe der Systemzeit als Zeichenkette
25	INITA	Initialisierung der Tastatur und der Systemuhr
26	SETDM	Setzen der Pufferadresse für Kassette lesen/schreiben
27	GETM	Abfrage der Speicherkonfiguration
28	SETM	Setzen der Speicherkonfiguration
29	DCU	Löschen Cursor
30	SCU	Setzen Cursor
31	COEXT	Zeichenkette vorverarbeiten
32	BOSER	nicht genutzt
33	RRAND	Lesen eines Blockes von Kassette (vgl. READS)

2.2.1. Beschreibung der Systemrufe

INIT C=0

Funktion:

- volle Initialisierung des Computers
- Übergang in den OS-Mode

CONSI C=1

- Funktion: - Eingabe eines Zeichen vom aktuellen CONST-Gerät
 - kein automatisches Echo auf dem Bildschirm

Return:

- A Zeichencode
- CY Fehlerstatus

Beispiel:

;Warten auf Betätigung der ENTER-Taste

```

      LD    C,1
M1:   CALL 5
      JPC  FEHL      ;Fehler bei Zeicheneingabe
      CMP  0DH      ;Code für ENTER
      JRNZ M1-#
  
```

CONSO C=2

Funktion:

- Ausgabe eines Zeichens zum aktuellen CONST-Gerät

Eingang:

- E Zeichencode

Return:

- CY Fehlerstatus

Beispiel:

;Löschen Bildschirm in Hintergrundfarbe blau

```
LD    C,2
LD    E,15          ;Farbsteuercode Hintergrund
CALL  CBOS
LD    E,4           ;Farbe BLAU
CALL  CBOS
LD    E,0CH         ;Code für CLEAR SCREEN
CALL  CBOS
```

```
....
CBOS: CALL 5
      RNC          ;kein Fehler
;Fehlerbehandlung
```

READI C=3

Funktion:

- Eingabe eines Zeichens vom aktuellen READER-Gerät

Return:

- A Zeichencode
- CY Fehlerstatus

PUNO C=4

Funktion:

- Ausgabe eines Zeichens zum aktuellen PUNCH-Gerät

Eingang:

- E Zeichencode

Return:

- CY Fehlerstatus

LISTO C=5

Funktion:

- Ausgabe eines Zeichens zum aktuellen LIST-Gerät

Eingang:

- E Zeichencode

Return:

- CY Fehlerstatus

GETST C=6

Funktion:

- Abfrage der Spielbebel

Return:

- B Spielhebel 2
- C Spielhebel 1

- Bitbelegung des B- bzw. C-Registers:

Bit	Richtung Spielhebel 1	äquivalente Taste (nur für Spielhebel 1)
0	links	←
1	rechts	→
2	runter	↓
3	hoch	↑
4	Aktionstaste	ESC oder Komma

beliebige Kombinationen sind möglich

Beispiel:

;Warten auf Betätigung der Aktionstaste 1

```

M1: LD    C,6
    CALL  5
    LD    A,C
    AND   10H      ;Ausblenden aller anderen Kombinationen
    JRZ   M1-#
  
```

GETIO C=7

Funktion:

- Abfrage des I/O-Bytes

Return:

- A I/O-Byte

SETIO C=8

Funktion:

- Setzen des I/O-Byte

Eingang:

- E I/O-Byte

PRNST C=9

Funktion:

- Ausgabe einer Zeichenkette zum aktuellen CONST-Gerät
- die Zeichenkette kann beliebige SteuerCodes zur Zeichenausgabe enthalten (siehe 2.3.2.)
- das Ende der Zeichenkette ist ein Byte 00

Eingang:

- DE Adresse der Zeichenkette

Return:

- CY Fehlerstatus

Beispiel:

;Löschen Bildschirm in Hintergrundfarbe blau

;Ausgabe einer Kopfzeile in der Farbe rot

```

LD    C,9
LD    DE,TEXT
CALL  5
    . . .
  
```

;Zeichenkettendefinition

TEXT: DB 15H ;Farbsteuercode Hintergrund

```

DB      4           ;Farbe BLAU
DB      0CH         ;Code für CLEAR SCREEN
DB      14H         ;Farbsteuercode Vordergrund
DB      1           ;Farbe ROT
DB      'Kopfzeile'
DA      0A0DH       ;CRLF
DB      0           ;Ende der Zeichenkette

```

RCONB C=10

Funktion:

- Eingabe einer Zeichenkette vom aktuellen CONST-Gerät
- die Zeichenkette wird in einem Pufferbereich abgelegt, dessen Länge vom Nutzer zu initialisieren ist
- das die Zeichenkette abschließende ENTER wird nicht im Puffer abgelegt
- die Zeichenkette wird vom BOS automatisch abgeschlossen, wenn das Pufferende erreicht wurde (kein versehentliches Überschreiben der folgenden Speicherbereiche.
- die Tasten DEL und ← löschen das Zeichen vor dem Cursor
- die Taste CLLN löscht die gesamte Zeichenkette
- die Tasten ↑ und ↓ werden ignoriert
- das Zuschalten des Druckers (CTRL/P) ist möglich; die interne Codierung (10H) wird jedoch nicht in die Zeichenkette übernommen
- alle anderen SteuerCodes sind verwendbar und werden in die Zeichenkette übernommen (siehe 2.3.2.)
- Pufferaufbau

LPU	LZK	1.Z	2.Z	3.Z
-----	-----	-----	-----	-----	-------

```

LPU  Länge des Puffers vom Nutzer initialisiert
LZK  Länge der Zeichenkette (vom BOS aktualisiert)
1.Z  1. Zeichen der Zeichenkette
2.Z  2. Zeichen der Zeichenkette

```

Eingang:

- DE Adresse des Zeichenpuffers

Return:

- CY Fehlerstatus
- bei Eingabe von STOP wird die Zeichenkette abgebrochen, die Rückkehr erfolgt mit CY=1 und Fehlercode=0

Beispiel:

;Anfordern einer Eingabe

```

LD      C,9
LD      DE, TEXT
CALL    CBOS           ;Ausgabe der Anforderung
LD      C,10
LD      DE,PUFFE
LD      A,40
LD      (DE),A         ;initialisieren max. Zeichenzahl
CALL    CBOS
JPC     STOP           ;STOP-Taste gedrückt
INC     DE
LD      A,(DE)         ;lesen eingegebene Zeichenzahl
OR      A
JPZ     LEER           ;leere Zeichenkette (nur ENTER)

```

;Auswertung

```

CBOS:   ...
CALL    5
RNC           ;kein Fehler

```

```

        OR    A
        SCF
        RZ                                ;Fehlerstatus durch STOP-Taste
;Fehlerbehandlung
        ....
;Definitionen
TEXT:  DA    0A0D                        ;CRLF
        DB    'Ihre Eingabe'
        DB    'bitte'
        DB    0
PUFFE: BER    1                        ;Länge Puffer
        BER    1                        ;Länge Zeichenkette
        BER    80                       ;max. Zeichenzahl

```

CSTS C=11

Funktion:

- Abfrage Status des aktuellen CONST-Gerätes

Return:

- A Status
 keine Taste betätigt A = 0
 Taste betätigt A = Zeichencode
- CY Fehlerstatus

RETVN C=12

Funktion:

- Abfrage der Versionsnummer des Monitors (für eventuelle Modifikationen des Monitors)

Return:

- BC Versionsnummer

OPENR C=13

Funktion:

- Ausgabe der Ausschrift 'start tape'
- Lesen Block 0 einer Datei von Kassette
- Vergleich gelesener Dateiname mit gesuchtem Dateinamen
- bei Namensgleichheit übernehmen der gelesenen Dateiparameter in den FCB (siehe 2.3.4.)
- Ausgabe eines Leerzeichens zum aktuellen CONST-Gerät
- Kassettenpuffer ist die aktuelle DMA-Adresse (siehe Ruf 26)

Eingang:

- Name und Typ der gewünschten Datei im FCB
- DMA (1BH) Adresse Kassettenpuffer für Block 0

Return:

- aktualisierte Dateiparameter im FCB (Anfangsadresse, Endadresse, Startadresse, Schutzbyte)
- LBLNR (6CH) nächste zu lesende Blocknummer (1)
- CY Fehlerstatus

CLOSR C=14

Funktion:

- beenden Kassette lesen
- mit diesem Ruf wird kein Block von der Kassette gelesen

Return:

- BC Adresse der aktuellen Dateiparameter (Anfangsadresse, Endadresse, Startadresse, Schutz-byte)

OPENV C=15

Funktion:

- Ausgabe der Ausschrift 'start tape'
- Ausgabe von Block 0 auf Kassette

Eingang:

- Name Typ und Dateiparameter im FCB (von Nutzer zu initialisieren) (siehe 2.3.4.)

Return:

- A Nummer des geschriebenen Blocks (0)
- BLNR Blocknummer des nächsten Blocks (1)
- CY Fehlerstatus

CLOSW C=16

Funktion:

- Ausgabe des letzten Blockes einer Datei auf Kassette

Return:

- A Nummer des geschriebenen Blockes (FF)
- BLNR die Merzkelle der Blocknummer hat den Wert 0
- CY Fehlerstatus

GETCU C=17

Funktion:

- gleichzeitige Abfrage der logischen und physischen Cursoradresse

Return:

- D Zeile des Cursors (1-24)
- E Spalte des Cursors (1-40)
- BC physische Cursoradresse
- CY Fehlerstatus

Hinweis:

- bei Änderung des Monitors mit Verlegung des BOS-Stackbereichs durch den Nutzer, ist dieser Ruf nicht mehr verwendbar

SETCU C=18

Funktion:

- Setzen des Cursors durch eine logische Adressierung

Eingang:

- D Zelle (1-24)
- E Spalte (1-40)

Return:

- CY Fehlerstatus

READS C=20

Funktion:

- Lesen eines Blockes einer Datei von der Kassette
- Ausgabe eines Leerzeichens zum aktuellen CONST-Gerät

Eingang:

- LBLNR (6CH) zu lesende Blocknummer
- DMA (1BH) Adresse, auf welcher der Block abgelegt wird

Return:

- A Kennzeichen für letzten Block der Datei (EOF)
 kein EOF 0
 EOF 1
- LBLNR LBLNR neu = LBLNR alt + 1
- DMA DMA neu = DMA alt + 80H

- CY Fehlerstatus
- BLNR die wirklich gelesene Blocknummer (auch im Fehlerfall)

Hinweis:

- das Lesen kann an beliebiger Stelle der Datei beginnen
- solange die gelesene Blocknummer kleiner ist als die gesuchte, liest das Programm weiter
- ist die gelesene Blocknummer größer, kehrt das Programm mit Fehler 'record not found' zurück
- wird der letzte Block erkannt, so wird dieser Block eingelesen und das Programm kehrt mit A=1 (EOF) zurück

Beispiel:

;einfaches Programm zum Lesen einer Datei

;der FCB (5CH) wurde bereits mit dem Namen der gewünschten Datei belegt

```
LD    DE,80H            ;Standard-Kassettenpuffer
LD    C,26            ;Ruf SETDM
CALL   CBOS
LD    C,13            ;Ruf OPENR
CALL   CBOS
LD    DE,(ANFAD)    ;gelesene Anfangsadresse der Datei (6DH)
LD    C,26            ;nach DMA
CALL   CBOS
LD    C,20
M1:   CALL   CBOS
      OR    A
      JRZ   M1-#       ;weiterlesen bis EOF
```

;Endebehandlung

```
....
CBOS CALL   5
      RNC            ;kein Fehler
```

;Fehlerbehandlung

....

WRITS C=21

Funktion:

- Schreiben eines Blockes einer Datei auf Kassette

Eingang:

- BLNR (6BH) Nummer des zu schreibenden Blockes
- DMA (1BH) Speicheradresse, ab der zu schreiben ist

Return:

- A Nummer des geschriebenen Blockes
- BLNR BLNR neu = BLNR alt + 1
- DMA DMA neu = DMA alt + 80H

SETTI C=22

Funktion:

- Stellen der internen Uhr

Eingang:

- A Stunde
- D Minute
- E Sekunde

Hinweis:

- die Uhr wird für die Dauer von Kassettenlese- und Kassettschreiboperationen unterbrochen

GETTI C=23

Funktion:

- Abfrage des aktuellen Standes der internen Uhr

Return:

- A Stunde
- D Minute
- E Sekunde

PRITI C=24

Funktion:

- Ausgabe des aktuellen Standes der internen Uhr als Zeichenkette
- die Zeichenkette ist 8 Bytes lang und nicht durch ein Byte 0 abgeschlossen
- Form der Zeichenkette Stunde: Minute: Sekunde

Eingang:

- DE Adresse des Speicherbereiches für die Zeichenkette (Adresse kann auch im Bildwiederholpeicher liegen)

Beispiel:

;Zeitanzeige während des Tastaturpollings

;

;Hauptprogramm

```
    . . . .  
    CALL  EINGZ      ;Eingabe eines Zeichens  
    . . . .
```

;

;Zeicheneingaberoutine

```
EINGZ: LD      C,24  
        LD      DE,POS      ;Bildschirmposition der Zeitausgabe  
        CALL    5  
        LD      C,11        ;Sys.-Ruf CSTS  
        CALL    5  
        OR      A  
        JRZ     EINGZ-#     ;noch keine Taste betätigt  
        LD      C,1  
        JMP     5           ;Zeichen abholen
```

Beispiel 2:

;Zeitanzeige durch Nutzung des Sekundeninterruptes des CTC

;Nutzer dieses Programms müssen gewährleisten, daß die Adressen 100H, , 10AH nicht durch ihr Programm verwendet werden

;(durch Ruf 24 überschrieben)

;Initialisierung

```
INITO: LD      HL,(206H)    ;Adresse Interrupt CTC 3  
        LD      (INTAD),HL
```

```

        LD    HL,TOUT      ;Startadresse der Zeitausgabe
        DI
        LD    (206H),HL
        EI
        RET
INTAD:  DA    0
;Zeitausgabe
TOUT:   PUSH  BC
        PUSH  DE
        PUSH  HL
        LD    HL,TOUT1
        PUSH  HL          ;Rückkehradresse bereitstellen
        LD    HL,(INTAB)
        JMP   (HL)        ;Ansprung der Uhrinterruptroutine
TOUT1:  LD    DE,POS
        LD    C,24
        CALL  5
        POP   HL
        POP   DE
        POP   BC
        RET

```

INITA C=25

Funktion:

- Initialisierung der Tastatur
- Initialisierung des Sekundentaktes der internen Uhr (keine Initialisierung der Uhrzeit)

SETDMC=26

Funktion:

- Setzen der Pufferadresse für Kassette lesen und schreiben

Eingang:

- DE Adresse des Kassettenpuffers

GETM C=27

Funktion:

- Abfrage der Speicherkonfiguration
- logischer Speichertest (Speicher ist in 1k Blöcken konfigurierbar)

Eingang:

- DE zu testende Adresse

Return:

- A Speicherstatus des 1k Bereiches, welcher die zu testende Adresse beinhaltet
 - 1 RAM beliebig nutzbar
 - 0 ROM oder kein Speicher oder
RAM bei Verwendung von Systemrufen
Schutz vor Überschreiben und Kopieren mittels Kassetten-E/A

Beispiel:

;Test auf vorhandenen Farbzusatzmodul

```

        LD    DE,0E800H   ;Adresse des Farbattributspeichers
        LD    C,27
        CALL  5
        OR    A

```

JPNZ FARBE
JMP SW ;nur schwarz/weiß

Hinweis:

- nach RESET oder Systemruf 0, ist die logische Speicherkonfiguration der physischen gleich
- ein Test der Adresse 1000H liefert das gleiche Ergebnis, wie der Test der Adresse 13FFH
- in beiden Fällen wird der Bereich 1000H bis 13FFH getestet

SETM C=28

Funktion:

Eingang:

- DE Adresse
- A Status
- 1 frei
- 0 geschützt

Beispiel:

;Schutz eines Programm es vor Überschreiben durch Nachladen von Kassette

```
PANF: LD HL,PANF
      LD DE,400H
      LD BE,PEND
      XOR A ;Status geschützt
M1:   EX DE,HL
      PUSH BC
      LD C,28
      CALL 5
      POP BC
      EX DE,HL
      ADD HL,DE ;nächste Adresse bereitstellen
      PUSH HL
      SBC HL,BC
      POP HL
      JRC M1-# ;Adresse <=PEND, weiter
      . . . .
PEND: EQU #
```

DCU C=29

Funktion: - Löschen des Cursors auf dem aktuellen CONST-Gerät

Hinweis:

- jede Zeichenausgabe bringt den Cursor sofort wieder zu Anzeige

SCU C=30

Funktion:

- Anzeige des Cursors auf dem aktuellen CONST-Gerät

Return:

- BC physische Cursoradresse

COEXT C=31

Funktion:

- Vorverarbeiten einer eingegebenen Zeichenkette

- Entfernen aller Steuercodes aus der Zeichenkette
- Anfügen eines abschließenden Bytes 00

Eingang:

- DE Adresse des Zeichenkettenpuffers (Pufferaufbau analog Systemruf 10)

Return:

- neue Zeichenkettenlänge und Zeichenkette im Zeichenkettenpuffer
- CY Zeichenkettenstatus
 - 1 zu bearbeitende oder resultierende Zeichenkette hat die Länge 0
 - 0 sonst

Beispiel:

;Eingabe und Vorverarbeitung einer Parameterzeile

```
LD   DE,ZKPU      ;Zeichenkettenpuffer
LD   C,10          ;Eingabe Zeichenkette
CALL CBOS
LD   C,31
CALL 5
JPC  NOPAR         ;keine Parameter, Zeile war leer
```

;Parameteranalyse

```
....
CBOS CALL 5
RNC                      ;kein Fehler
OR    A
JRNZ  FAUSW-#           ;Fehlerauswertung
INC   DE                ;STOP wurde gegeben
LD    (DE), A           ;löschen Puffer
DEC   DE
RET
```

RRAND C=33

Funktion:

- Lesen eines einzelnen Blockes einer Datei
- keine Veränderung von LBLNR und DMA (vgl. Systemruf 20)

Eingang:

- LBLNR (6CH) zu lesende Blocknummer
- DMA (1BH) Adresse auf welcher der Block abgelegt wird

Return:

- A Kennzeichen für letzten Block der Datei (EOF)
 - kein EOF 0
 - EOF 1
- CY Fehlerstatus
- BLNR (6BH) die wirklich gelesene Blocknummer (auch im Fehlerfall)

Hinweis:

- das Lesen kann an beliebiger Stelle der Datei beginnen
- solange die gelesene Blocknummer kleiner ist als die gesuchte, liest das Programm weiter
- ist die gelesene Blocknummer größer, so kehrt das Programm mit Fehler 'record not found' zurück
- wird der letzte Block erkannt, so wird dieser Block eingelesen und das Programm kehrt mit A=1 (EOF) zurück

2.2.3. Fehlerbehandlung

Der Monitor des Computers Z 9001 besitzt eine eigene Fehlerbehandlung. Diese wird bei Verwendung von Systemrufen aktiviert und gibt auf das aktuelle CONST-Gerät Fehlermeldungen aus. Ein Fehlerzustand wird durch das gesetzte CY-Flag angezeigt. Zusätzlich enthält das A-Register in diesem

Fall einen speziellen Fehlercode zur näheren Bestimmung des Fehlers. Fehlerzustand und Fehlercode werden dem Nutzer durch BOS übergeben.

Tabelle der Fehlercodes:

Fehler- code	Fehlermeldung	Fehlerursache
00	-----	dieser Code dient nur als Warnung(vgl. Sys.-Ruf 10)
01	error 1	Parameterfehler
02	error 2	Eingabefehler
03	error 3	Parameter außerhalb zulässiger Grenzen
04	error 4	Fehler bei Gerätezuweisung
05		frei für Erweiterung
06		frei für Erweiterung
07	BOS-error: OS	unzulässiger Systemruf
08	BOS-error: CONST	Fehler im Gerät CONST
	BOS-error: READER	Fehler im Gerät READER
	BOS-error: PUNCH	Fehler im Gerät PUNCH
	BOS-error: LIST	Fehler im Gerät LIST
09	BOS-error: memory protected	Speicherbereich ist geschützt
10	BOS-error: end of memory	logisches Speicherende erreicht
11	BOS-error: record not found	gelesene Blocknummer zu groß
12	BOS-error: bad record	Lesefehler
13	BOS-error: file not found	falscher Dateiname wurde gelesen

2.3. BASIC INPUT/OUTPUT SYSTEM (BIOS)

2.3.1. Überblick

Im Monitor des Z 9001 besteht für den Anwender die Möglichkeit, verschiedene Systemprogramme unter Umgehung des BOS direkt aufzurufen. Bei Verwendung dieser Direktrufe werden keine Register gerettet und keine Fehlermeldungen ausgegeben. Die Direktrufe belasten den Stack des Anwenders. Bestimmte Direktrufe erwarten Parameter oder übergeben Parameter an den Anwender.

Eingabeparameter:

- Wortparameter im BC - Register
- Byteparameter im C - Register

Ausgabeparameter:

- Wortparameter Im BC - Register
- Byteparameter im A - Register

Ein Fehler bei der Behandlung eines Direktrufes wird durch das gesetzte CY-Flag und einen Fehlercode im A - Register angezeigt (siehe 2.2.3.). Der Aufruf erfolgt über eine Sprungtabelle ab F000H.

BIOS-Sprungtabelle

Adresse	Funktion		
F000	JMP	INIT	Kaltstart (RESET)
F003	JMP	WBOOT	Warmstart (Teilinitialisierung)
F006	JMP	CONST	Abfrage Status CONST
			Return: A - Status
			0 keine Taste sonst Zeichencode
F009	JMP	CONIN	Eingabe Zeichen von CONST
			Return: A - Zeichen
F00C	JMP	COOUT	Ausgabe Zeichen zu CONST
			Eingang: C - Zeichen
F00F	JMP	LIST	Ausgabe Zeichen zu LIST

F012	JMP	PUNCH	Eingang: C - Zeichen Ausgabe Zeichen zu PUNCH
F015	JMP	READER	Eingang: C - Zeichen Eingabe Zeichen von READER
F018	JMP	GSTIK	Return: A - Zeichen Abfrage Spielhebel
			Return: C - Spielhebel 1 B - Spielhebel 2
F01B	JMP	BOSER	nicht genutzt
F01E	JMP	STIME	Stellen Systemuhr
			Eingang: A - Stunde C - Minute B - Sekunde
F021	JMP	GTIME	Abfrage Systemuhr
			Return: A - Stunde C - Minute B - Sekunde
F024	JMP	SDMA	Setzen Adresse Kassettenpuffer
			Eingang BC - Adresse
F027	JMP	READ	Lesen eines Blockes von Kassette
			Eingang: LBLNR - gewünschte Blocknummer
			Return: LBLNR neu = LBLNR alt + 1 DMA neu = DMA alt + 80H
F02A	JMP	WRITE	Schreiben eines Blockes auf Kassette
			Eingang: BLNR - zu schreibender Block
			Return: BLNR neu = BLNR alt + 1 DMA neu = DMA alt + 80H
F02D	JMP	LLIST	Abfrage Status LIST
			Return: A - Status
F030	JMP	GCURS	Abfrage physische Cursoradresse
			Return: BC - Adresse
F033	JMP	SCURS	Setzen physische Cursoradresse
			Eingang BC - Adresse
F036	JMP	BOSER	nicht genutzt
F039	JMP	GIOBY	Abfrage I/O-Byte
			Return: A - I/O-Byte
F03C	JMP	SIOBY	Setzen I/O-Byte
			Eingang: C - I/O-Byte
F03F	JMP	GMEM	Abfrage logische Speicherkonfiguration
			Eingang: BC - Adresse
			Return: A - Status 1 RAM 0 sonst
F042	JMP	SMEM	Setzen logische Speicherkonfiguration
			Eingang: BC - Adresse
			A - Status 1 RAM 0 sonst

Hinweis:

Direktrufe sollten nur für Programme genutzt werden, die mit dem installierten Monitor ohne Systemerweiterung arbeiten.

2.3.2. Spezielle Steuercodes

Der Computer Z 9001 ist in der Lage, spezielle Steuercodes zu erzeugen und zu verarbeiten. Diese dienen insbesondere der Farbsteuerung des Bildschirms. Die ordnungsgemäße Arbeit des Computers mit diesen Steuercodes ist jedoch nur bei Verwendung der im Monitor integrierten Standardtreiber für Tastatur und Bildschirm gewährleistet. Die folgende Tabelle gibt einen Überblick über alle verwend-

baren SteuerCodes des Z 9001. Ist bei der Beschreibung der Funktion der Wirkungsbereich des SteuerCodes nicht explizit angegeben, so sind diese Codes im OS-Mode, im Systemruf 2, 9 und 10 gleichermaßen verwendbar.

Hexcode	Funkt.- taste	CTRL- taste	Funktion
01		CTRL/A	
02	CLLN	CTRL/B	Sys.-Ruf 10 und OS-Mode: Löschen aller eingegebenen Zeichen
03	STOP	CTRL/C	Sys.-Ruf 10: Abbruch der Eingabe OS-Mode: Durchführen eines Warmstarts
04		CTRL/D	
05 (F)		CTRL/E	nächstes Zeichen ist Code für Bildschirmrandfarbe
06 (F)		CTRL/F	Blinken EIN/AUS für alle folgenden Zeichen
07		CTRL/G	Ausgabe eines Summertones
08	←	CTRL/H	Sys.-Ruf 2 und 9: Cursor nach links ohne Zeichenlöschen Sys.-Ruf 10 und OS-Mode: Löschen des letzten Zeichens
09	→	CTRL/I	Cursor nach rechts ohne Zeichenlöschen
0A	↓	CTRL/J	Sys.-Ruf 2 und 9: Cursor runter ohne Zeichenlöschen Bildschirm rollt aufwärts
0B	↑	CTRL/K	Sys.-Ruf 2 und 9: Cursor hoch ohne Zeichenlöschen Bildschirm rollt abwärts
0C		CTRL/L	Bildschirm löschen
0D	ENTER	CTRL/M	Sys.-Ruf 2 und 9: Cursor an den Anfang der Zeile ohne Zeichenlöschen Sys.-Ruf 10 und OS-Mode: Zeilenabschluß und Ausgabe CRLF
0E		CTRL/N	
0F		CTRL/O	
10		CTRL/P	Parallelausgabe auf LIST-Gerät EIN/AUS (LIST-Gerät muß vorher zugewiesen werden)
11		CTRL/Q	Kontrollton EIN/AUS
12		CTRL/R	
13	PAUSE	CTRL/S	
14 (F)	COLOR	CTRL/T	nächstes Zeichen ist Code für Vordergrundfarbe
15 (F)	COLOR+SHIFT	CTRL/U	nächstes Zeichen ist Code für Hintergrundfarbe
16 (F)		CTRL/V	Inversdarstellung aller folgenden Zeichen
17		CTRL/W	
18		CTRL/X	
19		CTRL/Y	
1A	INS		
1B	ESC		
1C	LIST		
1D	RUN		
1E	CONT		
1F	DEL		

(F) Wirkt nur bei eingebautem Farbmodul

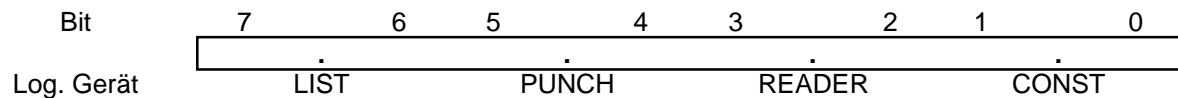
Die Tasten SHIFT, SHLOK, CTRL und GRAFIC erzeugen keine externen Codes. Die internen Farb-codes sind 0 (schwarz), 1 (rot), . . . und 7 (weiß).

2.2.3. Das I/O-Byte

Das I/O-Byte dient zur Verbindung von logischen und physischen Geräten. Für jedes der 4 logischen Geräte sind 4 physische Geräte wählbar. Die Zuschaltung eines dieser Geräte erfolgt über Änderung des I/O-Bytes. Zuvor müssen alle zu verwendenden Treiber zugewiesen und initialisiert werden.

Das I/O-Byte ist in 4 Bereiche zu je 2 Bits unterteilt. Jeder Bereich, der einem logischen Gerät zugeordnet ist, kann einen Wert von 0 bis 3 beinhalten und damit logisches und physisches Gerät verbinden.

Aufbau des I/O-Bytes:



Vom Monitor werden folgende Belegungen realisiert

CONST: 1 CRT ist aktiviert
 2 BAT ist aktiviert

LIST: 1 CRT ist aktiviert

Die Treiberadressen der jeweiligen physischen Geräte liegen auf den Adressen EFC9H bis EFE8H (siehe 1.2.2.).

2.3.4. Der FCB

Der FCB wird zur Beschreibung von Dateien auf einem externen Speicher (Kassette) verwendet. Vor jedem Auslagern einer Datei muß der FCB initialisiert werden. Mit dem Systemruf 15 (OPENW) wird dieser FCB als Block 0 auf Kassette geschrieben, um die Datei bei erneutem Einlesen eindeutig identifizieren zu können. Zum Einlesen ist der Systemruf 13 (OPENR) zu verwenden. Die Anfangsadresse des FCBs ist 5CH.

Aufbau des FCBs:

Adresse	Inhalt	Bemerkung
5CH	Dateiname	8 Zeichen mit Buchstaben beginnend ggf. mit 00 auffüllen
64H	Dateityp	3 Zeichen mit Buchstaben beginnend ggf. mit 00 auffüllen
67H	-	frei für Erweiterungen
68H	-	frei für Erweiterungen
69H	PSUM	Blockprüfsumme
6AH	ARB	interne Arbeitszelle
6BH	BLNR	Nummer zu schreibender Block oder Nummer gelesener Block
6CH	LBLNR	Nummer zu lesender Block
6DH	AADR	Anfangsadresse der Datei
6FH	EADR	Endadresse der Datei
71H	SADR	Startadresse eines Programms
73H	SBY	Schutzbyte

Hinweise:

- vor OPENW sind durch den Anwender Dateiname, Dateityp, Anfangsadresse, Endadresse, Startadresse und Schutzbyte zu initialisieren
- die Startadresse einer Datendatei oder eines nicht selbst startenden Programmes auf FFFFH setzen (kein versehentliches Starten bei Einlesen im OS-Mode, da Sprung auf RET-Befehl)
- vor OPENR sind Dateiname und Dateityp zu initialisieren
- nach OPENR sind Anfangsadresse, Endadresse, Startadresse und Schutzbyte durch die von Kassette gelesenen aktualisiert (nicht bei 'file not found')
- wird eine Datei mit Schutzbyte = 1 eingelesen, ist kein Schreiben auf Kassette mehr möglich

2.3.5. Kassettenaufzeichnung

Die Aufzeichnung von Programmen und Daten auf Kassette erfolgt in geblockter Form zu je 128 Bytes. Die Übertragungsrate beträgt etwa 1800 Baud. Zur Aufzeichnung werden 3 unterschiedliche Frequenzen verwendet.

0 - Bit	:	2500 Hz
1 - Bit	:	1250 Hz
Trennzeichen	:	625 Hz

Jedes Zeichen wird durch eine volle Periode aufgezeichnet.

a) Dateiaufbau:

Block 0	Dateikopf Inhalt ist der komplette FCB (siehe 2.3.4.)
Block 1, ..., n	Daten der Datei
Block FFH	letzter Datenblock der Datei Endeblock

b) Blockaufbau:

Vorton	Folge von 1 - Bit mit abschließendem Trennzeichen
Block 0:	6000
sonst	160
Blocknummer	1 Byte
Daten	128 Bytes
Prüfsumme	1 Byte (durch Addition aller Datenbytes gewonnen)

c) Byteaufbau:

Information	8 Zeichen
Trennzeichen	1 Zeichen

3. Zusätzliche Treiber

Der Monitor des Z 9001 bietet dem Anwender über das ASGN-Kommando die Möglichkeit, den logischen Geräten jeweils 4 physische Geräte mit beliebigen Treiberroutinen zuzuweisen. Um die Zusammenarbeit mit allen anderen Systemkomponenten zu gewährleisten, sind dabei besondere Randbedingungen zu beachten.

Treibereingangsparameter:

- A	Kommando (siehe 3.2.) (für CONST, READER, LIST)
- C	Zeichen (bei Kommando AUSGABE)
- DE	Adresse bei Cursorkommandos

Treiberausgangsparameter:

- A	Zeichen (bei Kommando EINGABE)
- CY	Fehlerstatus
- HL,DE	Adresse bei Cursorkommandos

Hinweis:

Vordergrundregister müssen bei Eintritt in das Treiberprogramm nicht gerettet werden (erfolgt bereits im BOS)

3.1. Treiberinitialisierung

Die Initialisierung sollte grundsätzlich über das ASGN-Kommando erfolgen. Eine automatische Initialisierung über die Nutzung des Kommandos (siehe 2.1.2.) und Direktzugriff auf entsprechende Systemzellen ist zu vermeiden.

Der Aufruf der Initialisierung im ASGN-Kommando erfolgt in der gleichen Weise wie der Start eines Anwenderprogrammes. Dabei kann es sich sowohl um ein transientes Kommando, als auch um ein Programm auf Kassette handeln. Ladbare Treiberprogramme auf Kassette werden durch das ASGN-Kommando selbständig eingelesen und zur Initialisierung gestartet. Derartige Programme sind in der Initialisierung mittels geeigneter RELOCATING-Routinen an das Ende des logischen RAM-Bereiches zu verschieben. Notwendiger Arbeitsspeicher ist gleichfalls dort einzurichten. Nach dem Verschieben ist der Zeiger des logischen RAM-Endes EOR (36H) auf eine Adresse vor dem Treiberprogramm zu stellen. Für jede Treiberoutine eines physischen Gerätes stehen im System-RAM (ab 280H) 4 Bytes Speicher zur Verfügung. Die Treiberinitialisierung muß der ASGN-Routine folgende Parameter übergeben:

- CY Fehlerstatus nach Initialisierung
- H logische Gerätenummer
 - 0 CONST
 - 2 READER
 - 4 PUNCH
 - 6 LIST
- L physische Gerätenummer (0, 1 3)
- BC Adresse der E/A-Routine
- DE Adresse einer Zeichenkette (wird nach der Zuweisung durch ASGN angezeigt)

Beispiel:

;PROM-Transientkommandoprogramm zur Initialisierung eines Druckers als TTY-Gerät

```
      JMP  INIT
      DB  'DRUCKER '
      DA  0
INIT:  LD   HL,(EOR)           ;Adresse end of RAM (36H)
      LD   DE,RAME-RAMA      ;benötigter Arbeitsspeicher
      SBC  HL,DE
      LD   (EOR),HL          ;neue log. RAM-Ende
      INC  HL
      LD   (RTTYL),HL         ;Zeiger auf Arbeitsspeicher (2B0H)
```

;Hardwareinitialisierung

```
      ....
      LD   H,6                ;log. Gerätenummer (LIST)
      LD   L,0                ;phy. Gerätenummer (TTY)
      LD   BC,OUTZ            ;Adresse der Zeichenausgabe
      LD   DE,TEXT
      OR   A                  ;kein Fehler
      RET
```

```
      ;
FEHL: SCF                    ;Fehler
      RET
```

```
      ;
;Zeichenausgabe
```

```
OUTZ: ....
```

```
      ;
;RAM-Definitionen
```

```
RAMA: ....
```

```
      ....
RAME: EQU  #
```

Hinweis:

- TTY-Treiber können beliebigen logischen Geräten zugewiesen werden
- CRT-Treiber können CONST und LIST zugewiesen werden
- alle anderen nur dem im H-Register angegebenen logischen Gerät (sonst Fehlermeldung im ASGN-Kommando)

3.2. Besonderheiten der Treiber für CONST

An die Treiber der für CONST möglichen Geräte (TTY, CRT, BAT, UC) werden besondere Anforderungen gestellt, da sie Routinen zur Eingabe, Ausgabe, Statusabfrage und Cursorsteuerung beinhalten müssen. Dem Treiber wird die zu erfüllende Aufgabe durch spezielle Kommandocodes im A - Register übergeben.

Kommandocodetabelle:

Code	Funktion

00	Abfrage Status Return: A Status 0 kein Zeichen bei Eingabegerät, nicht bereit bei Ausgabegerät sonst Zeichen liegt an bei Eingabegerät, (im installierten CRT-Treiber wird der Zeichencode übergeben)
01	Eingabe Zeichen Return: A Zeichen
02	Ausgabe Zeichen Eingang: C Zeichen
03	Cursor löschen
04	Cursor anzeigen Return: HL physische Cursoradresse
05	Abfrage logische und physische Cursoradresse Return: HL physische Cursoradresse DE logische Cursoradresse
06	Setzen Cursor auf logische Adresse Eingang: DE logische Cursoradresse
07	Abfrage physische Cursoradresse Return: HL physische Cursoradresse
08	Setzen Cursor auf physische Adresse Eingang: DE physische Cursoradresse
FF	Initialisieren/Rücksetzen des Gerätes

Hinweis:

- werden verschiedene Kommandos durch den Anwender nicht benötigt, kann dafür ein Sprung zum Fehlerausgang des Treibers programmiert werden
- korrespondierende System- und Direkturfe sind dann nicht mehr verwendbar

Beispiel:

;Eingangsverteiler für CRT-Treiber mit allen Funktionen

```
INC  A
JPZ  RESET      ;Initialisieren/Rücksetzen Gerät
DEC  A
JPZ  STAT       ;Status
DEC  A
JPZ  EING       ;Eingabe
DEC  A
JPZ  AUSG       ;Ausgabe
DEC  A
JPZ  LCUR       ;Löschen Cursor
DEC  A
```

	JPZ	ACUR	;Anzeige Cursor
	DEC	A	
	JPZ	ADRLC	;Abfrage log. und phy. Cursoradresse
	DEC	A	
	JPZ	SLOGC	;Setzen Cursor auf log. Adresse
	DEC	A	
	JPZ	ADRPC	;Abfrage phy. Cursoradresse
	DEC	A	
	JPZ	SPHYC	;Setzen Cursor auf phy. Adresse
FEHL:	SCF		;unzulässiges Kommando
	RET		;Fehlerausgang

3.3. Besonderheiten der Treiber für READER und LIST

Soll ein READER-Gerät Im Zusammenhang mit dem BATCH-Mode von CONST betrieben werden, so müssen im Treiber für das READER-Gerät (RDR, UR1, UR2) die Kommandos Eingabe, Statusabfrage und Initialisierung programmiert sein. Bei Treibern für das LIST-Gerät müssen die Kommandos Ausgabe, Statusabfrage und Initialisierung In jedem Falle programmiert werden.

- CDEL 2
 gerufen von: GVAL
 Parameter : A
 Z
 Test eines Zeichens auf Trennzeichen 0
 zu testendes Zeichen
 1

b) gerufene Programme
 - CDELI Zeichentest

c) Ausgang
 -

d) Return
 Parameter: A, C
 Z
 CY
 CONBU
 getestetes Zeichen
 0 kein Trennzeichen
 1 Trennzeichen
 1 Trennzeichen 0 (Kennzeichen für Stringende)
 getestetes Zeichen gelöscht mit Leerzeichen

Name : CHEC (F26CH)

Funktion: Stringvergleich

a) Eingang
 - CHEC
 gerufen von: LOCK, OPENR
 Parameter : HL
 DE
 B
 Adresse String 1
 Adresse String 2
 Anzahl zu vergleichender Zeichen

b) gerufene Programme
 -

c) Ausgang
 -

d) Return
 Parameter: Z
 DE
 HL
 1 String 1 = String 2
 unverändert
 Wort vor String 2

Name : CPROM (F28EH)

Funktion: Suchen Kommando im Speicher
 Kommando muß auf integraler 100H-Grenze beginnen

Aufbau: JMP KOMM
 DB 'KNAME ' ;8 Zeichen mit Space aufgefüllt
 DA 00 ;Ende Kommandofeld
 oder JMP KOMM 1
 DB 'K1NAME '
 DB 0 ;Ende Kommando 1
 JMP KOMM 2
 DB 'K2NAME '
 DB 0 ;Ende Kommando 2
 .
 .
 .
 JMP KOMM N '
 DB 'KNAME '
 DA 0 ;Ende Kommandofeld

a) Eingang

- CPROM
 gerufen von: WBOOT, GOCPM
 Parameter : INTLN INTLN+1 enthält Kommando in der Länge 8 Bytes

b) gerufene Programme

- LOCK Suchen String

c) Ausgang

d) Return

- | | | | |
|------------|----|---|-----------------------------------|
| Parameter: | Z | 0 | Kommando nicht gefunden |
| | | 1 | Kommando gefunden |
| | HL | | Adresse der Kommandoroutine |
| | DE | | Adresse des Kommandos im Speicher |

Name : DISPA (F0BDH)

Funktion: Anzeige der aktuellen Gerätezuweisung

a) Eingang

- DISPA
 gerufen von: ASGN, ALDEV
 Parameter : SADV Stringadreßvektor zur Adressierung der zum log. Gerät definierten Ausbestrings
 LOGDV Tabelle der logischen Geräte
- DISPE
 gerufen von: GOCPM Zeichen-E/A-Fehler in der Kommandoeingabe

b) gerufene Programme

- OCRLF Ausgabe CRLF
- PRNST Ausgabe String
- OUTA Ausgabe Zeichen

c) Ausgang

- WBOOT Fehler nach Gerätezuweisung bei Zeichenausgabe

d) Return

- Parameter: -

Name : EXIO (F2CEH)

Funktion: Lesen einer ausgewählten Treiberadresse
 Prüfen der gültigen Zuweisung

a) Eingang

- EXIO
 gerufen von: ALDEV, CONST
 Parameter : B interne Nummer des log. Gerätes (0, 2, 4, 6)
 0 : CONST
 6 : LIST
 IOBYT
 Treiberadreßvektor

b) gerufene Programme

- COMPW Vergleichen DE und HL

c) Ausgang

-

d) Return

Parameter:	CY	1 keine Treiberadresse gefunden (FFFFH)
		0 Adresse gefunden
	HL	Treiberadresse

Name : GVAL (F1EAH)

Funktion: Löschen internen Puffer (INTLN).
 Übernahme Parameter aus CONBU nach INTLN
 Test auf Parameterart
 Konvertieren Parameter, wenn dieser ein Wert ist

a) Eingang

- GVAL

gerufen von: WBOOT, GOCPM, ALDEV, STIME, LOAD
 Parameter : CONBU Konsolpuffer

b) gerufene Programme

- CDEL Übernahme Zeichen aus CONBU und Test
- CDEL1 Test Zeichen
- CDEL2 Test Zeichen
- CONV konvertieren Parameter

c) Ausgang

- ERINP Eingabefehler im Parameter

d) Return

Parameter:	Z	1 Parameter war Dezimalzahl
		0 Parameter war keine Zahl
	CY	0 kein Fehler
		1 Fehler im Parameter
	A	Konvertierte Dezimalzahl, wenn 2 = 1 und CY = 0
	C	den Parameter begrenzendes Trennzeichen
	B	Länge des Parameters
	HL	Adresse des nächsten Zeichens in CONBU
	CY'	0 kein weiterer Parameter in CONBU
		1 weitere Parameter
	A'	den Parameter begrenzendes Trennzeichen
	INTLN	Länge des Parameters
	INTLN+1. . .	übernommener Parameter
	CONBU	übernommener Parameter und Trennzeichen gelöscht mit Leerzeichen

4.2. Monitorroutinen im BOS/BIOS

Name : ERDIS (F5EAH)

Funktion: Ausgabe Fehlermeldung

a) Eingang

- ERDIS

gerufen von: BOSE, REA, GOCPM
 Parameter : A Fehlercode
 CY 1 (bei 0 RET)

b) gerufene Programme

- PRNST Ausgabe String
- OUTA Ausgabe Zeichen in A
- OCRCF Ausgabe CR/LF

c) Ausgang

-

d) Return

Parameter: CY 1
A Fehlercode

Name : GETMS (F35CH)

Funktion: Eingabe String in Monitorpuffer (80H)

a) Eingang

- GETMS
gerufen von: GOCPM, REQU
Parameter : -

b) gerufene Programme

- CONIN Eingabe Zeichen
- OUTA Ausgabe Zeichen in A

c) Ausgang

- RCONB
Parameter: DE Adresse Consolepuffer (80H)

Name : LOAD (F522H)

Funktion: Laden eines Programms

a) Eingang

- LOAD
gerufen von: INFIL, CLOAD
Parameter : CONBU mit gesuchtem Namen

b) gerufene Programme

- GVAL Parameterübernahme
- MOV verschieben Speicherbereich
- OPENR Eröffnen für Lesen
- READ Lesen eines Blockes
- REA Abfrage Bedienerhandlung bei Fehler

c) Ausgang

- ERPAR Parameterfehler
- ERINP Eingabefehler
- OCRLF Ausgabe CR/LF nach Einlesen
Parameter: CY 0 keine Fehler
1 Fehler, Code in A
Programm ab Anfangsadresse im Speicher
DMA nach Programm

Name : REQU

Funktion: Ausgabe String 'start tape' , warten auf ENTER

a) Eingang

- REQU
gerufen von: OPENR, OPENW
- REQUO

gerufen von: REA

b) gerufene Programme

- PRNST Ausgabe String
- GETMS Eingabe String

c) Ausgang

-

d) Return

Parameter: A FFH wenn STOP
 0 sonst

Name : PEA

Funktion: Ausgabe Fehlermeldung, warten auf Bedienerhandlung

a) Eingang

- REA
- gerufen von: LOAD
- Parameter: A Fehlercode

b) gerufene Programme

- REQUO warten auf Bedienerhandlung
- MOD verändern Speicherkonfiguration (bei Fehler 9)

c) Ausgang

-

d) Return

Parameter: CY 0 kein STOP
 1 STOP, Fehlercode in A

Name : **CONST, CONIN, CONOUT, LIST, LLIST, READER,**
 PUNCH (CONST1, LIST1, RDR1) (F756H)...

Funktion: Verzweigen zu log. Geräten

a) Eingang

gerufen von: Systemruf, Cursorruf

Parameter: bei OUT Zeichen in C

b) gerufene Programme

- EXIO Test Gerätezuweisung, Startadresse der Gerätetreiber holen

c) Ausgang

-

d) Return

Parameter: CY 0 kein Fehler

 1 Fehler, Code in A

 B interne log. Gerätenummer

 bei IN Zeichen

Name : **CRT** (F8F1H)

Funktion: Bildschirmtreiber, Tastaturtreiber, Steuerprogramm

a) Eingang

- CRT

gerufen von: CONST1 (über JMP (HL))

Parameter : A

Art des Rufes

FF Init.

0 Status Tastatur

1 Eingabe

2 Ausgabe

(C Zeichen bei Ausgabe)

3 Cursor löschen

4 Cursor setzen

5 Abfrage log. Cursoradr.

6 Setzen log. Cursoradr.

(DE Adresse; Zeile/Spalte)

7 Abfrage phys. Cursoradr.

8 Setzen phys. Cursoradr.

(DE Adresse)

b) gerufene Programme

- OC

Ausgabe ASCII-Zeichen

c) Ausgang

- INITA

Tastaturinitialisierung

- OCHAR

Ausgabe Zeichen

- OC

Ausgabe ASCII-Zeichen (für Cursoroperationen)

d) Return

Parameter: A
DE, HL

Zeichen bei IN

Adressen bei Cursorabfrage

DE log.

HL phys.

Name : DECK (FD33H)

Funktion: dekodieren Tastaturmatrix

a) Eingang

- DECOO

gerufen von: ICTC

b) gerufene Programme

- GPIOD

Abfrage Tastatur PIO

c) Ausgang

-

d) Return

Parameter: A
Z

Zeichen

1 Fehler

0 gültig

Name : FORMS (F836H)

Funktion: Formatieren String auf n Bytes für Konvertierung in interne Zahl
Ausblenden bzw. Einfügen von Vornullen

a) Eingang

- FORMS

gerufen von: VIEXT

Parameter:	DE A	Stringadresse neue gewünschte Länge
------------	---------	--

b) gerufene Programme
- MOVE verschieben Speicherbereich

c) Ausgang
-

d) Return
Parameter: DE
CY Adresse 1. Zeichen neuer String
0 kein Fehler
1 zu viele signifikante Stellen (keine Vornullen)

Name : OC (F97DH)

Funktion: phys. Bildschirmtreiber

a) Eingang
- OC
gerufen von: CRT, OCHAR
Parameter : C ASCII-Zeichen

b) gerufene Programme
- DELC Cursor löschen
- ROLU Rollen hoch
- ROLD Rollen runter
- MIAT Farbcode investieren

c) Ausgang
-

d) Return
Parameter: HL phys. Cursoradresse

Name : OCHAR (F88DH)

Funktion: Behandlung alle Sonderzeichen (Farbe, Bell, Blinken, Invers)

a) Eingang
- OCHAR
gerufen von: CRT
Parameter : C Zeichen

b) gerufene Programme
- COL Farbbehandlung
- BELL1 Tonausgabe vorbereiten/beenden
- INIVT Initialisieren Ton
- AUS1 Ausgabe Ton
- OC Ausgabe ASCII-Zeichen
- MIAT Farbcode invertieren

Name : ROLU, ROLS, ROLL, MOVE, DELLI (FA4FH)...

Funktion: Bildschirm rollen, letzte Zeile löschen

- a) Eingang
- ROLU, ROLD
gerufen von: OC
 - ROLL
gerufen von: ROLU, ROLD
Parameter : A 0 runter
 =0 hoch
- b) gerufene Programme
- MOVE Speicher verschieben

Name : VIEXT (F815H)

Funktion: Umwandlung Parameterstring (Dezimalzahl) in interne Darstellung (ein Byte)

- a) Eingang
- VIEXT
gerufen von: GEVAL
Parameter : DE Stringadresse
- b) gerufene Programme
- FORMS formatieren Eingabe-String
- c) Ausgang
-
- d) Return
- Parameter: DE Adresse nach String
 A Wert

4.3. Monitorroutinen zur Kassetten-E/A

Name : IBYTE (FFE8H)

Funktion: Lesen eines Bytes

- a) Eingang
- IBYTE
gerufen von: MAREK
- b) gerufene Programme
- LSTOP Lesen eines Bits
- c) Ausgang
-
- d) Return
- Parameter: A, E Byte
 CY 1 Fehler
 0 kein Fehler

Name : IKACT (FF43H)

Funktion: Interruptroutine Schreiben

- a) Eingang
- IKACT
Parameter: ARB Länge nächste Halbperiode

b) gerufene Programme

-

c) Ausgang

-

d) Return

Parameter: ARB 0

Name : IKEP (FFBDH)

Funktion: Interruptroutine lesen

a) Eingang

- IKEP

b) gerufene Programme

-

c) Ausgang

-

d) Return

Parameter A Länge einer Halbperiode

Name : KAUBT (FF18H)

Funktion: Ausgabe eines Bytes

a) Eingang

- KAUBT
gerufen von: KARAM
Parameter : A Byte
D Länge Halbperiode vorheriges Bit
- AUST
gerufen von: KARAM
Parameter : D Länge Halbperiode vorheriges Bit
- AUS1
gerufen von: KARAM
Parameter : D Länge Halbperiode vorheriges Bit

b) gerufene Programme

- AUS1, AUSO Ausgabe 1 Bit, 0 Bit
- DYNST Warten eine Halbperiode

c) Ausgang

-

d) Return

Parameter: D Länge Halbperiode vorheriges Bit

Name : KARAM (FED6H)

Funktion: Schreiben eines Blockes

a) Eingang

- KARAM
 - gerufen von: WRIT
 - Parameter : DMA Blockadresse
 - BLNR Blocknummer
 - BC Anzahl der Vortonzeichen

b) gerufene Programme

- INIC1 CTC vorinitialisieren
- INIVT Vorton initialisieren
- AUS1 Ausgabe 1 Bit
- AUST Ausgabe Trennzeichen
- KAUBT Ausgabe 1 Byte

c) Ausgang

- DYNST Ausgabe der letzten Halbperiode

d) Return

Name : LSTOP (FFD1H)

Funktion: Lesen eines Bits

a) Eingang

- LSTOP
 - gerufen von: MAREK, IBYTE
- LS1
 - gerufen von: MAREK

b) gerufene Programme

-

c) Ausgang

-

d) Return

- Parameter: C Länge der Periode
- CY 1 0 Bit
- 0 und C 90H Trennz.
- 0 und C - 90H 1 Bit

Name : MAREK (FF59H)

Funktion: Lesen eines Blockes

a) Eingang

- MAREK
 - gerufen von: RRAND
 - Parameter : DMA

b) gerufene Programme

- INIC1 CTC vorinitialisieren
- LSTOP Warten eine Periode
- LS1 Warten eine Halbperiode

- IBYTE

Lesen eines Bytes

c) Ausgang

-

d) Return

Parameter:

BLNR
PSUM
CY

Blocknummer
Prüfsumme
0 kein Fehler
1 Fehler