

TYPE 3080

SOFTWARE

1

Schnittstellen - Beschreibung von Bildschirm- und
Tastatur - Treiber

DATA COM
Ingenieurgesellschaft mbH
Altenwoogstraße 31
6750 KAISERSLAUTERN
Telefon 0631/40806 - Fs 45918

Für Bildschirm und Tastatur gibt es folgende Eingänge:

- I. Betrieb mit Funktionsnummer (Adresse FE9AH)
- II. Zeichenweiser Betrieb mit/ohne Steuerzeichenerkennung (Adresse FE09H/FE76H)
- III. Tastaturstatus abfragen (Adresse FE12H)
- IV. Warten auf Taste (Adresse FE73H)

I. Betrieb mit Funktionsnummer

Mit dieser Schnittstelle können folgende Funktionen ausgeführt werden:

0. Rückgabe des Parameters X von Funktion 1.

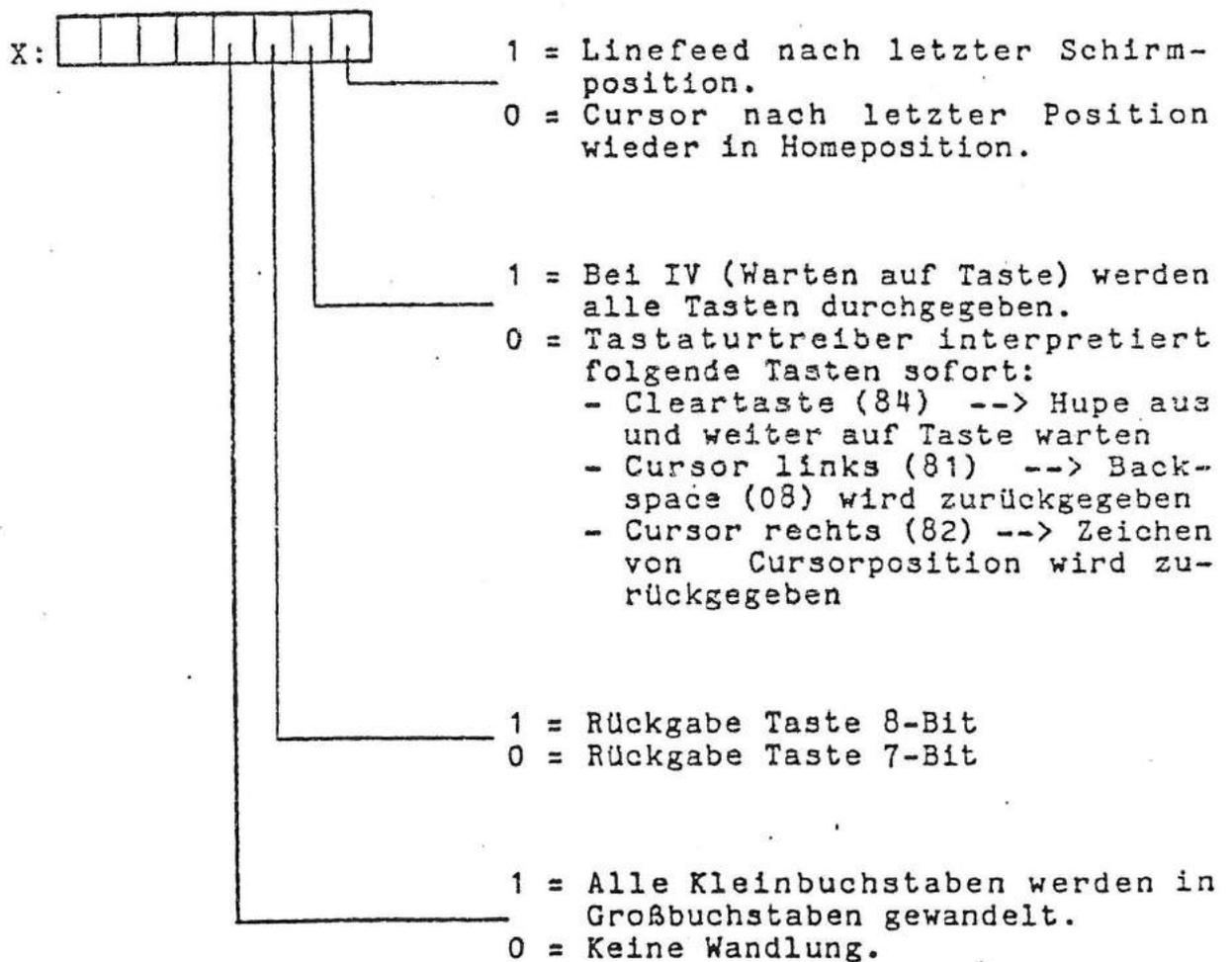
Aufruf: Reg. A = 0
Rückgabe: Reg. C = Parameter X

1. Generelle Initialisierung:

Aufruf: Reg. A = 1
Reg. E = Parameter X (Bedeutung siehe unten)

Der Bildschirm wird initialisiert, der Treiber wird grundgestellt (auch der Zentrier-Mode wird abgeschaltet, siehe Zentrieren), der Bildschirm wird gelöscht und der Cursor wird in die linke untere Ecke positioniert (letzte Zeile, erste Spalte).

Bedeutung des Parameters X:



Rückgabe: keine

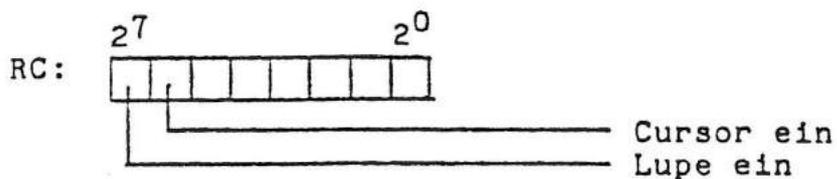
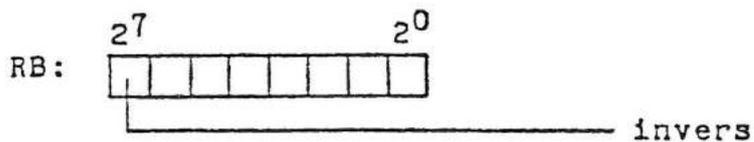
2. Display Status lesen

Es wird der Status des Display-Treibers, sowie die aktuelle Zeilen- und Spaltenanzahl zurückgegeben.

Aufruf: RA = 2

Rückgabe: RD = Anzahl Zeilen
RE = Anzahl Spalten

RB/RC = Status



Nur RH/RL sind unverändert.

3. Zeichen Lesen vom Bildschirm

Es wird das Zeichen der momentanen Cursorposition gelesen und der Cursor um eine Position weiterbewegt.

Aufruf: RA = 4

Rückgabe: RE = ASCII-Zeichen
RD = begleitende Bits (siehe Status).

4. Zeichen Schreiben auf Bildschirm

Das übergebene Zeichen wird an die momentane Cursorposition geschrieben und der Cursor um eine Position weiterbewegt.

Aufruf: RA = 5
RE = ASCII-Zeichen
RD = begleitende Bits (siehe Status).

Rückgabe: keine

5. Cursor-Position Lesen

Es wird die momentane Cursorposition zurückgegeben.

Aufruf: RA = 6

Rückgabe: RD = Zeilennummer
RE = Position in Zeile

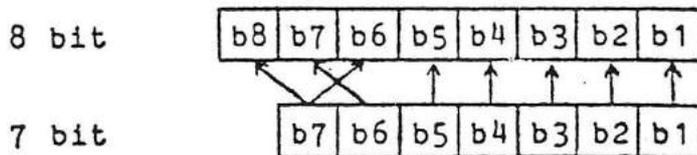
6. Tastaturtabelle laden

Es ist möglich, statt der im Betriebssystem implementierten Tastaturtabelle eigene Tastencodes zu verwenden. Die von der Taste gelieferte Matrix (siehe Layout-Schaubild) ist gleichzeitig der Index für die Tabelle. Dabei muß der Abstand zwischen Nicht-Shift-Code und Shift-Code 128 betragen.

Aufruf: Reg. A = 7
 Reg. H/L = Adresse Tastaturtabelle.

Nach dem Aufruf der Funktion steht der RAM - Bereich der neuen Tastaturtabelle wieder zur Verfügung. Wird in Register H/L 0000 übergeben, erfolgt eine Rückschaltung auf die im Betriebssystem implementierte Tastaturtabelle.

Die Tastaturtabellen sind im 8 Bit ASCII-Code abgelegt.



7. Umsetztabelle anhängen

Der Anwender kann der zeichenweisen Schnittstelle eine Umsetztabelle vorschalten. Dadurch ist es möglich, jedem beliebigen Code eine bestimmte Bildschirmfunktion zuzuordnen. Die Tabelle sieht folgendermassen aus:

	Code 1 Byte	Länge 1 Byte	String 6 Byte

insgesamt
16 Einträge

Aufruf: Reg. A = 8
 Reg. E = Anzahl der Einträge (maximal 16)
 Reg. H/L = Adresse der Tabelle

Nach dem Aufruf steht der RAM-Bereich der Tabelle wieder zur Verfügung.
 Wird in Register H/L 0000 übergeben, erfolgt ein "Abhängen" der Tabelle.

Für Sondertasten, die nicht vom Tastatur-Treiber durchgegeben sondern intern verarbeitet werden, sind folgende Codes reserviert:

SHIFT-Taste	C1H
SHIFT-Lock-Taste	COH
Wiederholtaste	C2H
Control-Taste	C5H

Bei allen Funktionen werden alle nicht genannten Register, mit Ausnahme des Registers A und des Bedingungsregisters, unverändert zurückgegeben.

II. Zeichenweiser Betrieb mit Steuerzeichenerkennung

Es wird jeweils 1 Zeichen übergeben (in Register C), wobei die unten aufgeführten Steuerzeichen interpretiert werden. Alle Register, mit Ausnahme des Registers A und des Bedingungsregisters, werden unverändert zurückgegeben. Bei einigen Steuerfunktionen muß ein Escape-Zeichen (1BH) vorausgehen.

Folgende Steuerzeichen werden wie beschrieben interpretiert:

07 BELL (BEL)

Die Clear-Lampe leuchtet auf und die Hupe ertönt. Beide Signale dauern ca. 1 Sek. an.

08 Cursor links (BS)

Der Cursor wandert um eine Stelle nach links.

0A Zeilenschaltung (LF)

Der Cursor wandert um eine Zeile nach unten. Ist der Cursor in der untersten Zeile wird das Bild um eine Zeile nach oben gerollt und die unterste Zeile gelöscht. Die Spaltenposition des Cursors bleibt dabei unverändert.

0C Grundstellung ohne Lupe

Der Bildschirmcontroller wird initialisiert, der Bildschirm-Treiber wird grundgestellt. Die Cursorposition ist in der linken unteren Ecke (letzte Zeile, erste Spalte). Der Bildschirm wird gelöscht. War Lupe eingeschaltet, bleibt dieser Mode erhalten, ebenso Zentriermode.

0D Ende Zeile

Der Cursor wird an den Anfang der Zeile gestellt.

11 Lupe einschalten

Lupe wird eingeschaltet, Bildschirm gelöscht, Cursor steht in der linken unteren Ecke. War Zentriermode eingeschaltet, wird dieser zurückgesetzt. Dergleichen alle Hintergrund-Modifikationen.

12 Endecode Zeichenmodifikation

Hintergrundbits werden grundgestellt.

13 Lupe ausschalten

Lupe wird ausgeschaltet, Bildschirm gelöscht, Cursor steht in der linken unteren Ecke. War Zentriermode eingeschaltet, wird dieser zurückgesetzt. Desgleichen alle Hintergrund-Modifikationen.

1C Invers einschalten

Alle folgenden Zeichen werden invers dargestellt.

Steuerzeichen mit Escape:

1B, 10, XX Blanks ausgeben

Es werden XX Blanks ausgegeben, der Cursor wandert entsprechend weiter.

1B, 11 Cursor home

Der Cursor wird in die obere linke Ecke gestellt (erste Zeile, erste Spalte).

1B, 12 Cursor ausschalten

Der Cursor wird ausgeschaltet, die Position jedoch weiterverwaltet.

1B, 13 Cursor einschalten

Der Cursor wird eingeschaltet.

1B, 14 ROLL-UP

Das Bild wird um eine Zeile nach oben gerollt. Die oberste Zeile wird unten angehängt. Die relative Cursorposition bleibt unverändert (Zeilen- und Spaltenposition bleiben gleich).

Diese Funktion sollte nicht in Verbindung mit der Zentrierfunktion angewendet werden.

1B, 15, XX, YY Zentrieren

Mit dieser Funktion kann eine erwünschte Schreibbreite mit XX (ungerade Zahl, da Spalten bei 0 beginnen), bzw. Schreibhöhe mit YY (gerade Zahl, da Zeilen mit 1 beginnen) eingestellt werden. Das Bild wird automatisch in die neue Bildmitte zentriert. Der Cursor wird in die neue Home-Position gebracht (nach wie vor Zeile 1, Spalte 0). Der Bildschirmteil außerhalb des gewählten "Fensters" ist dann nicht mehr zugänglich. Die Funktion Grundstellen (Formfeed 0C) läßt den zentrierten Zustand bestehen, d. h. der Cursor steht in der linken unteren Ecke des gewählten Bereiches. Es wird allerdings der gesamte Bildschirm gelöscht.

1B, 16, XX, YY Cursor positionieren

Der Cursor wird auf Zeile XX, Spalte YY positioniert.

1B, 17 Rest des Bildschirm löschen

Der Bildschirm wird ab der momentanen Cursorposition gelöscht. Die Cursorposition bleibt unverändert.

1B, 18 Rest der Zeile löschen

Die Zeile wird ab der momentanen Cursorposition gelöscht. Die Cursorposition verändert sich nicht.

1B, 19 Rest der Zeile löschen

Die Zeile wird ab der momentanen Cursorposition gelöscht, der Cursor wird auf den Anfang der nächsten Zeile positioniert.

1B, 1A Cursor rechts

Der Cursor wird um eine Position nach rechts bewegt.

1B, 1C Cursor up

Der Cursor wird um eine Zeile nach oben positioniert. Am oberen Bildrand erfolgt kein Roll-down.

1B, 1D, XX, YY

Es werden XX Zeichen mit dem Code YY ausgegeben.

1B, 1E, XX Hintergrund setzen

Es wird ab sofort jedes Zeichen mit dem angegebenen Hintergrund ausgegeben. Die Belegung ist identisch mit RB bei der Funktion Display-Status lesen.

1B, 1F, XX, YY Cursor positionieren mit Offset

Wie 1B, 16, XX, YY, jedoch müssen Zeile und Spalte mit einem Offset von 20 H übergeben werden.

Alle anderen Zeichen werden auf dem Bildschirm ausgegeben und der Cursor wird um eine Position weiterbewegt.

Die Schnittstelle ohne Steuerzeichenerkennung (Adresse FE76H) gibt alle Codes direkt auf den Bildschirm.

III. Tastaturstatus abfragen

Mit dieser Routine kann ermittelt werden, ob eine Taste angeschlagen wurde oder nicht.

Aufruf: keine Parameter

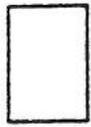
Rückgabe: RA = 0, ZERO = 1 => keine Taste
RA = FF, ZERO = 0 => Taste angeschlagen
CARRY = 1 => Übertragungsratenfehler
CARRY = 0 => kein Übertragungsratenfehler

IV. Warten auf Taste

Diese Routine wartet bis eine Taste angeschlagen wird, und gibt dann den Tastencode zurück (Besonderheiten siehe Generelle Initialisierung, Bedeutung Parameter X).

Aufruf: keine Parameter

Rückgabe: RA = Tastencode



00



01



02



03



04



05



06



07



08



09



0A



0B



0C



0D



0E



0F



10



11



12



13



14



15



16



17



18



19



1A



1B



1C



1D



1E



1F

ITT 3030

SOFTWARE

Schnittstellenbeschreibung

MOS - Anteile

für CP/M und BOS Betriebssysteme

Zeichenweiser Betrieb

Der Strom-Routine wird das jeweilige Zeichen übergeben bzw. von ihr übernommen. Um die Organisation der Daten im eigentlichen Gerät braucht sich der Benutzer nicht zu kümmern (z. B. Blockung usw.). Der zeichenweise Betrieb ist standardmäßig mit folgenden Treibern/Geräten möglich:

- System Ein/Ausgabegerät
- Tastatur
- Display
- Drucker

Bei Fehlern in den Routinen wird CARRY = 1 gesetzt. Die jeweilige Fehlerrückmeldung wird in den einzelnen Routinen genauer angegeben.

Schnittstelle für Command-Output und Lister

Funktion CO, LO

Aufruf: Ausgabezeichen RC
 mit Interpretation Sonderzeichen

Rückgabe: Meldung Fehler CARRY = 1
 Treiberidentifikation RA wenn CARRY = 1
 Status RB / RC wenn CARRY = 1

Bemerkung: Register RH / RL und RD / RE sind unverändert.
 Rückgabe Status siehe DROPEN

Schnittstelle für Command-Input

Funktion CI

Aufruf: -

Rückgabe: Eingabezeichen RA
 Meldung Fehler CARRY = 1
 Treiberidentifikation RA wenn CARRY = 1
 Status RB / RC wenn CARRY = 1

Bemerkung: Register RH / RL und RD / RE sind unverändert.

Man beachte, daß Status und Treiberidentifikation nur gültig sind, wenn CARRY gesetzt ist. Andernfalls sind beide undefiniert. Eine Randbedingungs-Fehlerbehandlung muß deshalb immer mit der CARRY - Abfrage ansetzen.

Schnittstelle für Reader-Input

Funktion RI

Aufruf: -

Rückgabe: Empfangenes Zeichen RA

Bemerkung: Alle Register sind verändert.
 Einstellung Baudrate über
 Funktion V24T Initialisieren.

Schnittstelle V24-Treiber

Funktion: V24T

1. Funktion Initialisieren (01)

Aufruf:	Funktionsnummer	RA
	Baudrate (Zählrate)	RD/RE
	Steuerbyte	RC

Rückgabe: keine

Tabelle Zählrate für Baudrate:

<u>Mode</u>	<u>Baud-Rate</u>	<u>Werte (Register RD/RE)</u>	
	4800	0807	7
	2400	080E	14
./ .16.	1200	081B	27
senden u.	600	0837	55
empfangen	300	086D	109
	110	4877	
./ . 1	19200	081B	
nur senden	9600	0837	

Steuerbyte RC:

Bit 2⁰ 0 = Wandlung 8 --> 7 Bit USASCII
 1 = Wandlung 7 --> 8 Bit USASCII

 Bit 2¹ 0 = ohne Parität
 1 = mit Parität

 Bit 2² 0 = mit gerader Parität
 1 = mit ungerader Parität

 Bit 2³ 0 = 1 Stopbit
 1 = 2 Stopbits

 Bit 2⁷ 0 = ./.. 16 Mode
 1 = ./.. 1 Mode

2. Funktion Empfangen (80)

Aufruf:	Funktionsnummer	RA
	Pufferadresse	RB/RC
	Länge	RD/RE
Rückgabe:	Länge	RD/RE

3. Funktion Senden (81)

Aufruf:	Funktionsnummer	RA
	Pufferadresse	RB/RC
	Länge	RD/RE
Rückgabe:	Länge	RD/RE

Switch-Byte

Es besteht die Möglichkeit alle CO - Ausgaben parallel auf dem Drucker auszugeben (List command output). Diese Funktion wird mit dem Switch-Byte gesteuert.

Bit 2^0 = 0 kein Ausdruck

Bit 2^0 = 1 Ausdruck

Funktion:	SWGET	Switch-Byte holen
	SWPUT	Switch-Byte ablegen

Aufruf:	bei SWPUT	Switch-Byte	RA
	bei SWGET	Switch-Byte	RA

Bemerkung: Alle anderen Register bleiben unverändert.

Tastatureingabe, Bildschirmanzeige und Statusabfrage

Funktion: CSTS

Aufruf: -

Rückgabe: keine Taste RA = 0, CARRY=0, ZERO=1
 Taste RA = OFFH, CARRY=0, ZERO=0
 Übertragungsratenfehler RA = OFFH, CARRY=1

Bemerkung: Bis auf RA sind alle Register unverändert. Die Taste wird nicht verarbeitet und muß mit CI abgeholt werden.

Eingabe mit gleichzeitiger Anzeige auf dem Bildschirm

Funktion: TI

Es wird auf ein Zeichen vom CI-Strom gewartet und übergeben mit gleichzeitigem Ausgeben des Zeichens auf den CO-Strom.

Aufruf: -

Rückgabe: Eingabezeichen RA
 Meldung CARRY=1 falls Fehler
 Treiberidentifikation RA wenn CARRY=1
 Status RB / RC wenn CARRY=1

Textstring ausgeben

Funktion: TXCO

Aufruf: Adresse String RH / RL

Rückgabe: -

Bemerkung: RH/RL und RD/RE sind unverändert. Es wird ein String über CO ausgegeben. Der String muß folgenden Aufbau haben:

1. Byte Länge der Ausgabe
2. Byte n Byte Ausgabedaten

Spezialfunktionen Display

Funktion: DISC

Es werden Spezialfunktionen des Displays aufgerufen. Die genaue Schnittstelle und die Bedeutung ist der Beschreibung über den Display-Kanaltreiber zu entnehmen.

Eingabe Hexadezimalwerte

Funktion: EXSTD

Mit Hilfe einer Steuertabelle wird eine Eingabe gesteuert.

Aufruf: Adresse Steuertabelle RH / RL

Aufbau Steuertabelle

1.	Byte	Anzahl einzugebende Werte
2. + 3.	Byte	Standardbelegung erster Eingabewert
4. + 5.	Byte	Standardbelegung zweiter Eingabewert
usw.		

Rückgabe: Im Stack eingegebene Werte, letzter Wert als oberster Wert.

Bemerkung: Alle Register sind unverändert.

Es werden soviele Werte eingegeben wie angegeben sind. Die einzelnen Werte sind durch Komma getrennt. Soll bei einem Wert der Standardwert genommen werden, so wird nichts eingegeben.

Beispiel:

Tabelle:	5, 1234H, 1, 2, 5678H, 8
Eingabe:	5, 6, 3
Übergebene Werte:	5, 6, 1, 3, 5678H, 8

Seite umschalten

Funktion: SWITCH
Aufruf: Seite in RC
Programmstartadresse in RH/RL
Rückgabe: -
Bemerkung: Register bleiben unverändert.
Rückschaltung muß vom gestarteten Programm
verwaltet werden.

Hardware - Erkennung

Funktion: HWERK
Aufruf: -
Rückgabe: Adresse eines Parameterblockes in H/L

Ausgabe Hexadezimalwert

Funktion: LBYTE
Aufruf: auszugebendes Byte RA
Rückgabe: -
Bemerkung: Register RH/RL und RD/RE sind unverändert.
Das Byte wird als zwei Zeichen über CO ausgegeben.

Ausgabe Hexadezimaladresse

Funktion: LADR
Aufruf: auszugebende Adresse RH/RL
Rückgabe: -
Bemerkung: Register RH/RL und RD/RE sind unverändert.
Die Adresse wird als vier Zeichen über CO ausgegeben.

Ausgabe CR, LF

Funktion: CRLF
Aufruf: -
Rückgabe: -
Bemerkung: Register RH/RL und RD/RE sind unverändert.
Es werden über CO die Zeichen CR (ODH) und LF (OAH) ausgegeben.

WandelroutinenASCII 7 Bit nach ASCII 8 Bit

Funktion: CODW78

Aufruf: zu wandelndes Zeichen RA

Rückgabe: gewandeltes Zeichen RA

Bemerkung: alle anderen Register bleiben unverändert.

ASCII 8 Bit nach ASCII 7 Bit

Funktion: CODW87

Aufruf: zu wandelndes Zeichen RA

Rückgabe: gewandeltes Zeichen RA

Bemerkung: alle anderen Register bleiben unverändert.

Hexadezimalzeichen nach ASCII - Zeichen

Funktion: CONV

Aufruf: zu wandelndes Zeichen RA

Rückgabe: ASCII - Zeichen RA und RC

Bemerkung: alle anderen Register bleiben unverändert.

Es werden nur die untersten vier Bit des Registers in ein ASCII - Zeichen gewandelt.

ASCII - Zeichen nach Hexadezimalwert

Funktion: NIBBLE

Aufruf: ASCII - Zeichen RA

Rückgabe: Hexawert in den untersten 4 Bit RA
CARRY = 0 gültiges Zeichen
CARRY = 1 kein Hexadezimalzeichen

Bemerkung: alle anderen Register bleiben unverändert.

HilfsroutinenRegister RH/RL + 1 und Vergleich mit Register RD/RE

Funktion: HILO
 Aufruf: Vergleichswerte RH/RL, RD/RE
 Rückgabe: -
 CARRY = 1 ZERO = 1 RH/RL = 0
 CARRY = 0 ZERO = 0 RH/RL < RD/RE
 CARRY = 0 ZERO = 1 RH/RL = RD/RE
 CARRY = 1 ZERO = 0 RH/RL > RD/RE
 Bemerkung: nur RA ist verändert.

Bereich mit Konstante füllen und Test ob richtig gefüllt

Funktion: FILLM
 Aufruf: Anfangsadresse RH/RL
 Endeadresse RD/RE
 Füllkonstante RC
 Rückgabe: Zelle nicht schreibbar CARRY
 Bemerkung: alle Register verändert.

Bereich verschieben

Funktion: MOVEM
 Aufruf: Quelladresse RH/RL
 Endeadresse RD/RE
 Zieladresse RB/RC
 Rückgabe: -
 Bemerkung: alle Register verändert.

Achtung: Bei Überlappung von Quell- und Zielbereich kann der Quellbereich zerstört werden. Es wird in aufsteigender Richtung umgeladen.

Stackpointer holen und merken

Funktion: SPGET

Aufruf: -

Rückgabe: Stack Pointer RH/RL

Bemerkung: außer RH/RL alle Register unverändert.

Speicherendeadresse laden

Funktion: MEMCK

Aufruf: -

Rückgabe: Endeadresse Anwender-Speicher RH/RL

Bemerkung: alle anderen Register unverändert.

Die Speicherendeadresse wird jeweils beim CP/M Kaltstart und Warmstart getestet.

Speicherendeadresse setzen

Funktion: MEMSET

Aufruf: Setzen Ende des Anwender-Speichers RH/RL

Rückgabe: -

Bemerkung: alle Register unverändert.

Random = Zahl = laden

Funktion: RNDZ

Aufruf: -

Rückgabe: Random Zahl RA

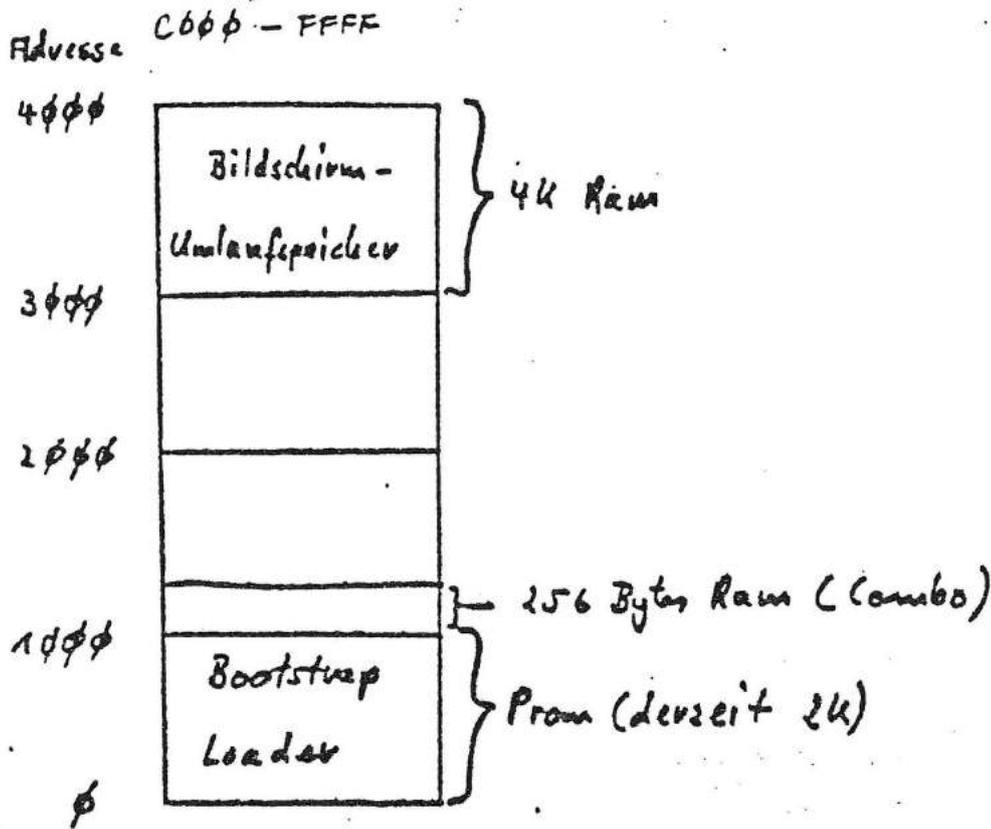
Bemerkung: alle anderen Register unverändert.

1. Tabelle der Funktionsadressen:

ADR	NAME	FUNKTION
FE00	SWITCH	Speicherumschaltung
FE03	CI	Command Input zeichenweise
FE06	RI	Reader Input zeichenweise
FE09	CO	Command Output zeichenweise
FE0C		Reserviert
FE0F	LO	Lister Output zeichenweise
FE12	CSTS	Consolstatus abfragen
FE15	HWERK	Rückgabe Hardware
FE18	TI	Console CI mit CO
FE1B	HILO	HL + 1 Vergleich mit DE
FE1E	TXCO	Textausgabe über CO
FE21		Reserviert
FE24		Reserviert
FE27	FILLM	Bereich füllen
FE2A		Reserviert
FE2E		Reserviert
FE32		Reserviert
FE36		Reserviert
FE3A	SWGET	Switch - Byte holen
FE3E	SWPUT	Switch - Byte setzen
FE42		Reserviert
FE46		Reserviert
FE4A	MEMCK	Speicheradresse holen
FE4E	MEMSET	Speicheradresse setzen
FE52	SPGET	Stack-Pointer holen und merken
FE55	EXSTD	Eingabe Hexadezimalwerte

ADR	NAME	FUNKTION
FE58		Reserviert
FE5B		Reserviert
FE5E	V24T	V24 Treiber
FE61		Reserviert
FE64	CODW78	Codewandlung 7 nach 8 bit
FE67	CODW87	Codewandlung 8 nach 7 bit
FE6A		Reserviert
FE6D		Reserviert
FE70	DROPEN	Einstellung Drucker
FE73	TASTE	Tastatur Treiber
FE76	DISPE	Display Treiber (Zeichen direkt a. Bildschirm)
FE79	DRUCE	Drucker Treiber
FE7C		Reserviert
FE7F	MOVEM	Bereich verschieben
FE82	LBYTE	Hexadezimal-Byte ausgeben
FE85	LADR	Hexadezimal-Adresse ausgeben
FE88	CONV	Hexadezimalwert nach ASCII-Zeichen
FE8B	NIBBLE	ASCII-Zeichen nach Hexadezimalwert
FE8E	CRLF	CR, LF auf CO ausgeben
FE91		Reserviert
FE94		Reserviert
FE9A	FAOR	Steuerfunktion Display <i>Reliert mit Fun Chon...</i>
FE9D	RNDZ	Random - Zahl lesen
FEA1	DENTRY	Eingang Floppytreiber
FEA4	SENTRY	Select-Eingang Floppytreiber

Speicher und 10-Aufteilung Seite 8



10 - Adressen

Combo - Chip		E0	-	EF
Wait - Zyklen	OP-Code	F0		
"	Memory	F1		
"	IO	F2		
DTR/RTS	setzen	F3		
DSR/CTS	lesen	F0		
Page	umschalten	F6		
Kombi - Interface I		20	-	3F
Floppy - Interface		50	-	5F
Winchester - Platte		10	-	1F

TYPE 3080

SOFTWARE

1

BESCHREIBUNG

Schnittstelle

Floppy - Treiber

- 0. Einleitung
- 1. Schnittstellen-Übersicht
 - 1.1 Klassifikation
 - 1.2 Aufbau des Treibereingangs
 - 1.3 Zwischenrückgabe
 - 1.4 Aufrufparameter
 - 1.5 Rückgabeparameter
- 2. Beschreibung der Treiberfunktionen
- 3. Datenorganisation auf einer Diskette
 - 3.1 Gliederung in Spuren und Sektoren
 - 3.2 Form eines Sektors

0. Einleitung

Der Floppy-Treiber ist ein Kanalprogramm zum Betrieb von maximal drei Doppelkopf-Laufwerken am SKS-MC80 System.

Die Steuerung des Treibers erfolgt über Aufruf- und Rückgabeparameter, welche in den Registern und Flags übergeben werden.

Der Floppy-Treiber ist nicht reentrant und normalerweise nicht unterbrechbar. Bei langdauernden Operationen unter Hardware-Kontrolle hat der Anwender die Möglichkeit der Zwischenrückgabe, um eigene Programme zwischenzeitlich abzuarbeiten.

1. Übersicht über Funktionen und Aufbau des Treibers

1.1 Treiberfunktionen

Folgende Funktionen lassen sich über den Treiber abwickeln

- Sektorweise Lesen und Schreiben im quasi - wahlfreien Zugriff zu Segmenten von 256 Bytes Länge (128 Bytes bei 8" Disketten).
- Schreib-/Lesekopfpositionierung spur- und sektorweise
- Initialisierung des Interfacecontrollers und der Laufwerke

1.2 Treibereingänge

Es existieren zwei Haupteingänge

1. Select Entry (SENTRY):

Laufwerksnummer und Diskettenseite werden in einem Parameter explizit angegeben.

2. Direct Entry (DENTRY):

Laufwerksnummer und Diskettenseite ändern sich gegenüber dem vorhergehenden Aufruf nicht.

1.3 Zwischenrückgabe Option

Einige Funktionen des Treibers dauern gegenüber den Programmlaufzeiten lang und laufen ausschliesslich unter Hardware- und Interface Kontrolle. (Überwiegend Positioniervorgänge).

In diesen Fällen wird der Treiber über eine Zwischenrückgabe verlassen.

Der Anwender hat dann die Möglichkeit, andere Programme abzuarbeiten (Hintergrundprogramme, Interruptserviceroutinen usw.).

Alle Register und Flags stehen zur Verfügung. Der Prozessor beginnt mit gesperrtem Interrupt.

Keine der Hardware-Funktionen ist zeitkritisch. Es liegt im Belieben des Anwenders, wie lange das Zwischenprogramm dauert.

Die Adresse des Zwischenprogramms muss mindestens einmal, bei Initialisierung des Treibers, übergeben werden, kann aber beliebig oft durch Neuinitialisierung geändert werden. Die Rückkehr in den Treiber erfolgt mit einem Return-Befehl (Rückkehradresse liegt im Stack!).

1.4 Allgemeine Aufrufparameter

Die Parameterübergabe bei Treiberaufruf geschieht funktions-spezifisch.

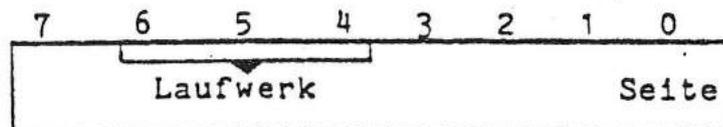
Die folgenden Parameter sind jedoch bei allen Funktionen gemeinsam und müssen in jedem Fall übergeben werden:

Eingang SENTRY: Funktionsnummer in RA
 Laufwerk, Seite in RL

Eingang DENTRY: Funktionsnummer in RA
 (Laufwerk, Seite wie bisher)

Die Funktionsnummern sind in Tabelle I zusammengefasst.

Der Parameter Laufwerk/Seite in RL
 hat die Form



Die funktions-spezifischen Parameter werden bei der Beschreibung der einzelnen Funktionen behandelt.

1.5 Allgemeine Rückgabeparameter

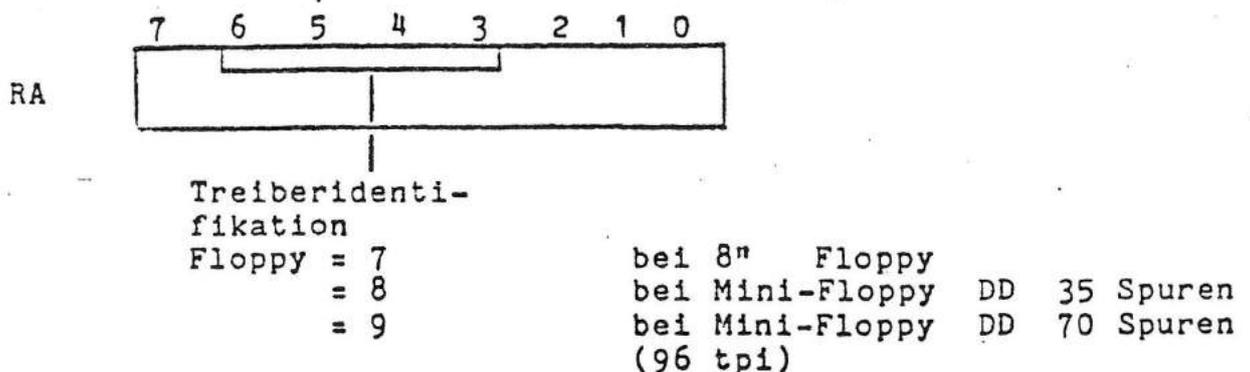
Folgende Parameter werden von allen Funktionen zurückgegeben:

Frühwarnung: ZERO = 1
 Fehler: CARRY = 1
 Status: in RB/RC (siehe Tabelle II)

Identifikation/Wiederholung in RA

Alle Register und Flags, ausser RH/RL werden verändert.

Der Parameter Identifikation hat die Form:



Anmerkung: Die Funktion "Treiberstatus lesen" (OF) bildet bezüglich Aufruf- und Rückgabeparameter Ausnahmen. Siehe Funktionsbeschreibungen.

Befehlsklasse IO		Befehlsklasse CTRL	
Nr.	Bit 2 ⁷ = 1	Nr.	Bit 2 ⁷ = 0
-	-	-	-
81	Sektorweise Schreiben bis zur angegebenen Pufferlänge (ohne read after write)	01	Initialisieren, Laufwerk/Spur selektieren
82	Sektorweise Lesen bis zur angegebenen Pufferlänge	-	-
83	Sektorweise Schreiben bis zur angegebenen Pufferlänge (mit read after write)	-	-
84	Positionieren auf Spur = TN; Sektor = SN (nur für formatierte Disk)	-	-
-	-	-	-
-	-	-	-
-	-	-	-
-	-	0A	Schreib-Lesekopf über Spur 0, Sektor 1 positionieren (BOD)
-	-	0C	Positionieren über Spur = TN, Sektor = 1 (für unformatierte Disketten)
-	-	0D	Spur formatieren
-	-	0E	Doorlock bedienen optional
-	-	0F	Treiberstatus lesen

Tabelle I: Funktionsnummern der Floppy Treiber

Reg. Bit	Bedeutung	Flag
	7 S/L-Kopf steht über letzter Spur, letzter Sektor	
	6 S/L-Kopf steht über erster Spur, erster Sektor	
RB	5 -	
	4* Laufwerk Interface unklar, Diskette fehlt	
	3 -	
	2* Schreibschutz	
	1 -	
	0 -	
	7 CRC - Fehler	SIGN
	6 Frühwarnung	ZERO
	5 -	
	4 Sektor nicht gefunden	
RC	3 -	
	2 -	
	1 Unzulässiger Vorgang oder Parameter	
	0 Fehlersummenmeldung (Anforderung nicht vollständig ausgeführt)	CARRY

Tabelle II Zusammenfassung der Statusrückmeldungen.

* siehe Funktion Treiberstatus lesen.

2. Beschreibung der Treiberfunktionen

2.1 Funktion Sektorweise Schreiben (81)

Es wird ab momentaner Position die im Aufruf angegebene Anzahl Datenbytes geschrieben. Der Schreib-Lesekopf steht nach dem Schreiben auf dem nächsten Sektoranfang. Es erfolgt kein Prüfllesen.

Aufruf:	Pufferadresse	RB/RC
	Anzahl Datenbytes	RD/RE
Rückgabe:	Position Sektor/Spur	RD/RE
	Schreibfehler	SIGN = 1, CARRY = 1

2.2 Funktion Sektorweise Lesen (82)

Es wird ab der momentanen Position die im Aufruf angegebene Anzahl Datenbytes gelesen. Der Schreib-Lesekopf steht nach dem Lesen auf dem nächsten Sektoranfang.

Aufruf:	Pufferadresse	RB/RC
	Anzahl Datenbytes	RD/RE
Rückgabe:	Position Sektor/Spur	RD/RE
	Lesefehler	SIGN = 1, CARRY = 1

2.3 Funktion Sektorweise Schreiben (83)

Wie Funktion 81, jedoch erfolgt anschliessend ein Prüfllesen.

2.4 Funktion absolutes Positionieren auf Spur und Sektor (84)

Der Schreib-Lesekopf wird auf die angegebene Position gebracht.

Aufruf:	Sektor	RD
	Spur	RE
Rückgabe:	Position Sektor/Spur	RD/RE

2.5 Funktion Initialisieren, Laufwerk/Spur selektieren (01)

Es wird die Zwischenrückgabeadresse eingestellt und an Laufwerk und Spur selektiert.

Aufruf:	Adresse eines Zwischenprogrammes	RB/RC
	Laufwerk/Seite (Belegung siehe 1.4)	RE

Rückgabe: siehe Allgemeine Rückgabeparameter (1.5).

2.6 Funktion Restore (0A)

Der Schreib-Lesekopf wird auf Spur 0, Sektor 1 positioniert.

Aufruf: siehe Allgemeine Aufrufparameter (1.4).

Rückgabe: siehe Allgemeine Rückgabeparameter (1.5).

2.7 Funktion Positionieren Spur (0C)

Es wird die angegebene Spur anpositioniert und das Sektorregister mit 1 geladen. Diese Funktion kann zum Positionieren unformatierter Disketten verwendet werden.

Aufruf: Spurnummer in RE

Rückgabe: siehe Allgemeine Rückgabeparameter (1.5).

2.8 Funktion Spur formatieren (0D)

Aufruf: RB/RC Beginn der Spurinformatio

Rückgabe: siehe Allgemeine Rückgabeparameter (1.5)

2.9 Doorlock bedienen - optional (0E)

Mit dieser Funktion kann in mit Doorlock ausgerüsteten Geräten dieser aktiviert bzw. deaktiviert werden.

Aufruf: RE = 00 Doorlock aus
RE = FF Doorlock ein

Rückgabe: siehe Allgemeine Rückgabeparameter

2.10 Funktion Treiberstatus lesen (0F)

Es wird der momentane Treiberstatus gelesen. Dabei werden nur die in Tabelle II mit * gekennzeichneten Bits zurückgemeldet.

Aufruf: siehe Allgemeine Aufrufparameter (1.4)

Rückgabe: RB (siehe Tabelle II)

3. Datenorganisation auf einer Diskette

3.1 Gliederung in Spuren und Sektoren

Eine Diskette hat 35 (77) Spuren zu je 16 (26) Sektoren.

1 Sektor hat eine Länge von 256 (128) Bytes.

Die Nummerierung der Spuren beginnt bei 0, die der Sektoren bei 1.

Die Zahlen in Klammern beziehen sich auf 8" Disketten.

Alle Ein/Ausgabeoperationen beginnen an der Sektorkante, die durch einen vorangegangenen IO- oder Steuerbefehl erreicht wurde.

3.2 Die Form eines Sektors

