

HANDBUCH fuer die  
INTEMEX 280 B / AP 22  
Karte mit 64 K Byte RAM

Installations- und Benutzungshinweise

Bitte ERST sorgfaeltig durchlesen,  
Dann die Karte benutzen

BRUNNEN 1911 314

## AP 22 Handbuch

Revision 1 - August 1983  
fuer BIOS Version 1.01

Copyright (C) 1983 bei IBS Computertechnik  
und Rainer Ellerbrake

### Verwendete Warenzeichen:

INTEL ist ein Warenzeichen der Firma INTEL Corp., USA

Apple II und Apple Pascal sind Warenzeichen der Firma Apple Computer, USA

CP/M ist ein eingetragenes Warenzeichen der Firma DIGITAL RESEARCH Inc., USA

Z80 ist ein eingetragenes Warenzeichen der Firma ZILOG, USA

SoftCard ist ein Warenzeichen der Firma Microsoft Consumer Products, USA

WORDSTAR ist ein Warenzeichen der Firma MicroPro International Corporation, USA

AM 22. JANUAR 1982

BEZUG: 1. 1. 1982  
ZUR: 21.01.1982

BEZUG: 1. 1. 1982  
ZUR: 21.01.1982

BEZUG: 1. 1. 1982

Inhaltsverzeichnis

<u>1.</u>	<u>Einleitung</u>	3
<u>2.</u>	<u>AP 22 Benutzerbeschreibung</u>	5
2.1	Unterstützung von Apple Peripheriekarten.	5
2.2	Disketten Laufwerke	7
2.3	Standard Ein-/Ausgabe	8
2.4	Installation der AP 22	9
2.5	Anpassung des SoftCard CP/M an die AP 22	9
2.6	Der Disk Manager (DM.COM).	13
2.7	Das Programm IBSCPM64.COM.	20
2.8	Das Selektiv-Loeschprogramm (ERAQ.COM)	21
2.9	Das Autoexecute Programm (AEXEC.COM)	22
2.10	Das Programm UCOPY.COM	24
<u>3.</u>	<u>Das IBS CP/M System</u>	25
3.1	Starten des Systems	25
3.2	Das Ein-/Ausgabe System	26
3.2.1	Logische und physikalische Gerate	26
3.2.2	Zuordnung der physikalischen Gerate	27
3.2.3	Unterstützte Peripheriekarten (ohne Disk Controller)	29
3.2.4	Eingabe Umcodierung	30
3.2.5	Ausgabe Umcodierung	34
3.2.5.1	Terminal unabhaengige Ausgabe	38
3.3	Disk Ein-/Ausgabe	43
3.3.1	Zuordnung der Disk Laufwerke	43
3.3.2	Unterstützte Disk Controller.	45
3.3.3	Die eingebaute Pseudodiskette.	47
3.4	BIOS Fehlermeldungen	48
3.5	Warmboot und Reset	51
3.6	Kommunikation Apple <-> AP 22.	52
3.6.1	RST1: 6502 Unterprogrammaufruf	53
3.6.2	RST2: Lesen einer 6502 Speicherzelle.	55
3.6.3	RST3: Schreiben in eine 6502 Speicherzelle	56
3.6.4	RST4: andere Funktionen	57
<u>4.</u>	<u>BIOS Details</u>	64
4.1	Arbeitsweise des BIOS.	64
4.2	Speicherbelegung	65
4.3	Konfigurationstabellen	70
4.4	Peripheriekarten Schnittstelle	72
4.5	Disketten Treiber Schnittstelle	74
<u>5.</u>	<u>Hardware Beschreibung</u>	84
5.1	Aufbau und Funktionsweise.	84
5.2	Ansprache der AP 22 durch den Apple IC	89
<u>6.</u>	<u>Kompatibilitaet zum SoftCard CP/M.</u>	92
6.1	Allgemeines	92
6.2	MBASIC und GBASIC.	95

Anhang

A Treiber fuer 80 Zeichen Karte, serielles und paralleles Interface  
B Ein/Ausgabe Umcodierung  
C Ein/Ausgabe Zuordnung  
D Disk Zuordnung  
E Peripheriekarten Definition  
F 6502 Mnemonics Makro Paket fuer MACRO-80  
G Belegung der Boot-Spuren

Tabellenverzeichnis

Tabelle 1: Zuordnung Logische -> physikalische Geraete. . . 26  
Tabelle 2: Geraetefunktionen . . . . . 28  
Tabelle 3: Standard Eingabe Umcodierungen . . . . . 31  
Tabelle 4: Standard Ausgabe Umcodierungen . . . . . 35  
Tabelle 5: Bildschirm Funktionen . . . . . 37  
Tabelle 6: AP 22 Speicherbelegung . . . . . 65-67  
Tabelle 7: 6502 Zero Page Belegung. . . . . 68  
Tabelle 8: Apple ][ Speicherbelegung . . . . . 69  
Tabelle 9: mittlere Zeitabweichung des 100 Hz Timers . . . 88  
Tabelle 10: AP 22 Speicherselektierung . . . . . 90

## 1. Einleitung

Bei der IBS AP 22 handelt es sich um eine Erweiterungskarte fuer Ihren Apple ][ Mikrocomputer. Sie verfuegt ueber 65536 x 8 bit (64K) Schreib-/Lesespeicher (RAM) und einen Z-80 B Mikroprozessor (CPU).

Dies bietet Ihnen einige zusaetzliche Moeglichkeiten:

1. Die mitgelieferten Programme ermoeglichen es, das CP/M Betriebssystem auf der AP 22 zu installieren. Dies ermoeglicht Ihnen den Zugriff auf eines der groessten existierenden Programmsortimente.
2. Der Schreib-/Lesespeicher kann unter den Betriebssystemen DOS 3.3 sowie Apple Pascal als Pseudo-Diskettenlaufwerk benutzt werden. Die Kapazitaet entspricht ungefaehr einer halben Apple ][ Diskette. Von Vorteil bei Benutzung dieser Moeglichkeit ist die kurze Datenzugriffszeit, nachteilig der Datenverlust bei Ausfall der Rechner-Betriebsspannung.

Zum Lieferumfang gehoeren:

- 1 Diskette AP 10/AP 21/AP 22 Pseudodisk Software fuer DOS 3.3 / SoftCard CP/M 2.20 / Apple Pascal 1.1
- 1 Diskette AP 22 - CP/M STARTER fuer DOS 3.3
- 1 Diskette AP 22 - CP/M PATCHER Z80 B Teil
- 1 AP 22 Handbuch mit Pseudodisk Beschreibung

sowie 1 AP 22 Karte

Bitte kontrollieren Sie bei Erhalt der Sendung sofort den Inhalt auf Vollstaendigkeit. Die CP/M Starter und die CP/M Patcher Diskette sind ab Version 1.01 zu einer Diskette zusammengefasst worden.

Nicht zum Lieferumfang gehoeren, aber ebenfalls benoetigt werden die SoftCard CP/M Master Diskette sowie die zugehoerige SoftCard Dokumentation. Diese muessen getrennt bezogen werden. Zur Anpassung eines neuen Disk Controllers benoetigen Sie zusaetzlich die im Abschnitt 4.5 angegebene Literatur.

Zweck und Aufbau dieser Beschreibung

Im Abschnitt 2 finden Sie alle Informationen, die zur Anpassung des Softcard CP/M an die AP 22 notwendig sind. Ausserdem sind dort auch die zusaetzlich mitgelieferten Programme beschrieben. Insbesondere die Abschnitte 2.4 und 2.5 sind dazu wichtig. Dieser Abschnitt ist auch fuer Benutzer mit geringer Erfahrung im Umgang mit CP/M gedacht.

Diese Beschreibung ist weder eine Einfuehrung noch eine Beschreibung des CP/M Betriebssystems, sie liefert lediglich AP 22 spezifische Informationen.

Im Abschnitt 3 finden Sie Informationen zur Anpassung des AP 22 BIOS, es kann die Zuordnung der Peripheriekarten zu den Slots veraendert werden, Eingabe "Funktionstasten" und Ausgabe Steuerzeichen koennen definiert werden, ausserdem finden Sie Informationen ueber BIOS Fehlermeldungen und die Moeglichkeiten zur Kommunikation mit dem 6502 ueber das BIOS der AP 22.

Der Abschnitt 4 ist fuer Systemprogrammierer mit guter Kenntnis des CP/M Betriebssystems ausgelegt und liefert Informationen zur Anpassung neuer Peripheriekarten an das AP 22 BIOS.

Im Abschnitt 5 wird die Hardware der AP 22 beschrieben, insbesondere auch die Ansprache der einzelnen Funktionen durch den Apple II.

## 2. AP 22 Benutzerbeschreibung

Die Installation der AP 22 ist einfach, Sie sollten aber einige Dinge vorher beachten, daher

Lesen Sie den nachfolgenden Abschnitt vor Installation der AP 22 sorgfaeltig durch!

Zur Installation benoetigen Sie auf jeden Fall einen Apple ][ plus, Apple ][e, ITT 2020 oder Basis 108 Rechner. Das AP 22 BIOS ist allerdings fuer den Apple ][ plus ausgelegt.

### 2.1 Unterstuetzung von Apple Peripheriekarten

Neben der AP 22 werden auch einige andere Apple Peripheriekarten unterstuetzt. Die einzelnen Slots des Apple ][ sind bestimmten Funktionen zugeordnet. Diese Zuordnung entspricht weitgehend der des SoftCard CP/M und des Apple Pascal Systems. Zu den einzelnen Funktionen gehoeren entsprechende Typen der Peripheriekarten. Diese sind im einzelnen:

<u>TYP</u>	<u>Karten-Name</u>
1	Apple Disk ][ Controller (16 Sektoren)
2	Apple Communications Interface oder aehnliche
3	Apple High Speed Serial Interface Apple Silentye Printer Interface Videx Videoterm 24 x 80 Terminal Card IBS AP 16-4 80 Zeichen Karte IBS AP 2 Serielles Interface IBS AP 4G Grafikfaehiges Druckerinterface IBS AP 21 dgl. mit eingebautem 56K Spooler IBS AP 11 Druckerinterface
4	Apple Parallel Printer Card
5	IBS AP 17 64 bis 256 K RAM Karte IBS AP 13a 64 K RAM Karte IBS AP 13b 128 K RAM Karte IBS AP 10 64 K RAM Karte mit 68B09 CPU (*) IBS AP 20 128 K RAM Karte mit 68000 CPU (*) IBS AP 21 64 K RAM Karte mit 6511 CPU (*) (**)

(\*) Die CPU wird nicht benutzt

(\*\*) Nur wenn die AP 21 nicht als Druckerspooler eingesetzt wird.

#### Hinweis:

Bei Speicherkarten mit 64K RAM kann es erforderlich sein, dass diese in einem Slot mit hoeherer Nummer stecken als die AP 22.

Die oben angegebenen Karten sind mit dem AP 22 CP/M System voll kompatibel und werden automatisch erkannt. Das heisst aber nicht, dass andere Peripheriekarten mit vergleichbaren Funktionen inkompatibel sind. Allgemein kann man sagen, dass unter Soft Card CP/M ohne Anpassungsprogramm lauffaehige Peripheriekarten ebenso unter AP 22 CP/M benutzt werden koennen. Ist diese Bedingung nicht erfuellt, so sind Aenderungen im Ein-/Ausgabesystem erforderlich.

Welche Peripheriekarten koennen in welchem Slot installiert werden? Aufschluss darueber gibt die nachfolgende Tabelle:

<u>SLOT</u>	<u>Karten-TYP</u>	<u>Funktion</u>
0	nicht benutzt	Dieser Slot kann z.B. die Language Card bzw. die IBS AP 1 enthalten; sie wird vom CP/M System als Pseudodisk(LaufwerkP:) Erweiterung benutzt
1	Typ 2, 3, 4	Drucker Interface (CP/M LST: Geraet)
2	Typ 2, 3, 4	Allgemeine Ein/Ausgabe (CP/M PUN: und RDR: Geraete)
3	Typ 2, 3, 4	Konsolen Ausgabe (CP/M CON: oder TTY: Geraet). Enthaelt normalerweise die 80-Zeichen Karte
4	Typ 1, 5	Disketten Laufwerk (CP/M Drive D:)
5	Typ 1, 5	Disketten Laufwerk (CP/M Drive C:)
6	Typ 1, 5	Disketten Laufwerk (CP/M Drive A: und B:) (*)
7	nicht benutzt	

(\*) Steckt in Slot 6 keine Disk Controller Karte, so wird das Boot- (Kaltstart-) Laufwerk als Drive A: benutzt.

Die AP 22 kann in einem der Slots 0 bis 6 installiert werden. Auf dem Apple IIe kann sie auch in Slot 7 verwendet werden (Jumper X7 muss offen bleiben). Die Zuordnung Funktion (->) Slot kann in den zugehoerigen Konfigurationstabellen geaendert werden.

## 2.2 Disketten Laufwerke

Wie im vorhergehenden Kapitel angegeben, koennen in den Slots 4, 5 und 6 Disk Controller Karten der Typen 1 und 5 installiert werden. Fuer die Start-Phase ist mindestens 1 Apple Disk II Controller in einem der Slots 1 bis 6 mit mindestens 1 Laufwerk erforderlich. Das Kaltstart- (Boot-) Laufwerk muss nicht als Laufwerk A: in Slot 6 installiert werden (dies ist allerdings i. d. R. zu empfehlen). Es wird nur in der Kaltstart-Phase, bei Betaetigen von <RESET> oder beim Nachladen des CP/M Systems benoetigt.

Bei den Disk Controller Karten des Typs 5 handelt es sich um Halbleiter-Speicher Pseudodisketten. Wird eine dieser Karten in einem Slot eingesetzt, der fuer 2 Laufwerke definiert ist, so wird jedem Laufwerk jeweils die Haelfte der Gesamtkapazitaet zugeordnet; ist nur 1 Laufwerk definiert, so erhaelt dieses die gesamte Kapazitaet.

### Beispiel:

In Slot 6 ist eine IBS AP 17 256 K RAM-Karte installiert. Es stehen 2 (Pseudo-) Laufwerke zur Verfuegung: Laufwerk A: mit 128K Byte und Laufwerk B: mit 128K Byte.

Wird die gleiche Karte in Slot 5 installiert, so steht nur 1 (Pseudo-) Laufwerk zur Verfuegung: Laufwerk C: mit 256 k Byte Kapazitaet.

Nachfolgend die Zuordnung der CP/M Laufwerke zu den Slots in Tabellenform

CP/M Laufwerk	Slot-Nr.	Laufwerk-Nr.
A:	6	1
B:	6	2
C:	5	1
D:	4	1

Neben diesen Laufwerken steht zusaetzlich eine eingebaute Speicher Pseudodiskette zur Verfuegung. Diese belegt den auf der Apple Mutterplatine vom CP/M-System ansonsten unbenutzten Speicherbereich. Sie kann als CP/M-Laufwerk P: angesprochen werden und hat eine Kapazitaet von 32 K Byte. Ist im Slot 0 eine "Language Card" installiert, so erhoehrt sich die Kapazitaet auf 48K Byte. Dies ist im Apple ][e und in der BASIS 108 immer der Fall.

### 2.3 Standard Ein- /Ausgabe

Die Slots 1 bis 3 sowie Apple Tastatur und Video-Teil werden fuer die Ein-/Ausgabe benutzt. Das CP/M Betriebssystem unterscheidet 4 logische Geraete. Diese sind:

#### 1. Die Console (CON:)

Nach dem Starten des Systems erfolgt ueber dieses Geraet die CP/M Kommando Eingabe und die System Ausgabe. Ist im Slot 3 eine Peripheriekarte der Typen 2, 3 oder 4 installiert, so erfolgt die Ein/Ausgabe ueber diese Karte, andernfalls ueber Apple Tastatur und 40 Zeichen Video Teil.

#### Hinweis:

Zur Unterstuetzung der Apple Tastatur werden einige eingegebene Zeichen in andere umgesetzt. Naeheres hierzu finden Sie im Kapitel "Eingabe Umcodierung".

#### 2. Die Drucker Ausgabe (LST:)

Ueber dieses Geraet koennen nur Ausgaben erfolgen. Ist im Slot 1 eine Peripheriekarte der Typen 2, 3 oder 4 installiert, so wird ueber diese Karte ausgegeben, andernfalls erscheinen die Daten auf dem 40 Zeichen Bildschirm.

#### Beispiel:

Ist weder im Slot 1 noch im Slot 3 eine Peripheriekarte installiert und wird mit CTRL-P das Mitprotokollieren auf dem LST: Geraet eingeschaltet, so erfolgen alle Ausgaben auf dem 40-Zeichen Bildschirm doppelt.

#### 3. "Leser" und "Stanzer" (RDR: und PUN:)

Diese Geraete waren urspruenglich zur Lochstreifen Ein/Ausgabe gedacht. Im AP 22 CP/M System koennen Eingaben (RDR:) ueber eine Peripheriekarte der Typen 2, 3 oder 4 in Slot 2 bzw. der Apple Tastatur und Ausgaben ueber die gleiche Peripheriekarte bzw. dem Apple 40-Zeichen Video-Teil erfolgen.

## 2.4 Installation der AP 22

Schalten Sie vor Beginn der Installation Ihren Apple aus!

Die AP 22 kann in einem der Slots 0 bis 6 installiert werden. Auf dem Apple IIc ist auch Slot 7 verfügbar. Vor dem Einsetzen der Karte kontrollieren Sie bitte den Slotstecker der AP 22 (die vergoldete Kontaktleiste). Sollten die Kontakte verschmutzt oder mit einer duennen klebrigen Schicht ueberzogen sein, so sind sie vorher zu reinigen. Anschliessend kann die AP 22 in den Slot eingesteckt werden. Manche Apple-kompatible Rechner erlauben es, die Karte zu tief einzustecken; die maximal zulaessige Tiefe ist erreicht, wenn der gruene Schutzlack ueberhalb der Kontakte mit der Oberkante des Slot-Steckers eine Linie bildet.

Damit ist die Installation der AP 22 beendet. Ist die AP 22 richtig installiert, so darf seitliches Wackeln im Betrieb nicht zu Funktionsstoerungen fuehren.

## 2.5 Anpassung des SoftCard CP/M an die AP 22

Im Lieferumfang der AP 22 ist das CP/M Betriebssystem nicht enthalten, es wird nur das notwendige Ein-/Ausgabe System (BIOS) mitgeliefert. Um ein lauffaehiges System zu erhalten, benoetigen Sie neben den mitgelieferten Programmen die Master-Diskette des SoftCard CP/M Systems der Firma Microsoft auf einer 16 Sektor Apple IIc Diskette. Diese Diskette kann mit einem 44K oder aber einem 56K CP/M System versehen sein.

Verwenden Sie zur Anpassung des Systems eine Kopie dieser Diskette (eine Kopie kann mit dem Programm COPYA unter dem Betriebssystem Apple DOS 3.3 erstellt werden).

### 1. Benoetigte Hardware

Ein Minimalsystem zur Anpassung besteht aus:

Apple IIc Plus mit 48 K RAM oder dazu kompatibelem Rechner, AP 22 in einem der Slots 0 bis 6 und einem Disk IIc Controller mit mindestens 1 Laufwerk.

Vom AP 22 CP/M System unterstuetzte Peripheriekarten duerfen zusaetzlich in den vorgesehenen Slots installiert sein, andere Karten, mit Ausnahme einer evtl. vorhandenen "Language Card" in Slot 0 oder einer "Farbkarte" in Slot 7, sollten aus dem Rechner entfernt werden.

Hinweis: Das AP 22 CP/M System arbeitet mit Programmunterbrechungen der 6502 CPU ueber das IRQ-Signal. Dieses Signal darf von anderen Peripheriekarten nicht benutzt werden!

### 2. Anpassung

Zur Anpassung benoetigen Sie folgende Disketten:

1 Diskette mit DOS 3.3 Betriebssystem,  
 die mitgelieferte "AP 22 - CP/M STARTER DOS 3.3" Diskette,  
 die mitgelieferte "AP 22 - CP/M PATCHER Z80B TEIL" Diskette sowie  
 eine Kopie der SoftCard Master Diskette.

Nachfolgend wird die Anpassung beschrieben. Auf der linken Seite  
 finden Sie die von Ihnen einzugebenden Texte fett gedruckt,  
 rechts die dazu gehoerenden Kommentare. Das Zeichen [ erhalten  
 Sie durch Betaetigen von <ESC> und A.

Legen Sie eine Diskette mit DOS 3.3 Betriebssystem ein und  
 schalten Sie Ihren Rechner ein.

DOS 3.3 meldet sich mit:

Legen Sie jetzt die "AP 22 - CP/M STARTER" Diskette ein. Befindet  
 sich auf Ihrer SoftCard Master Diskette ein 44 K CP/M System, so  
 geben Sie

**JRUN CP/M START** ein, fuer ein 56 k CP/M System:

**JRUN CP/M 56 START** Das Programm meldet sich mit:

**xxk CP/M START PROGRAMM** Jetzt wird das Ein/Ausgabe  
 System geladen, anschliessend  
 erfolgt die Anfrage:

**BITTE LEGEN SIE EINE KOPIE  
 IHRER xxk SOFTCARD MASTER  
 DISKETTE EIN  
 DISKETTE EINGELEGT (J/N)?**

Legen Sie jetzt Ihre SoftCard Master Diskette ein und quittieren  
 Sie die Anfrage mit J. Das CP/M System muss sich mit A $\times$  melden.  
 Ist dies nicht der Fall, so schalten Sie bitte Ihren Rechner aus  
 und pruefen ob die AP 22 und andere Peripheriekarten korrekt  
 installiert sind. Hat sich das System richtig gemeldet, so haben  
 Sie bereits ein lauffaehiges CP/M System, das aber nicht die  
 gesamten 64 K Byte RAM der AP 22 ausnutzt. Um dies zu erreichen,  
 sind die folgenden Anpassungen notwendig:

**A>PIP** CP/M Datentransferprogramm PIP  
 starten  
 \* PIP meldet sich

Legen Sie jetzt die "AP 22 - CP/M PATCHER" Diskette ein.

**\*P:=PATCHER.COM** Das Anpassungsprogramm wird in  
 der internen Pseudodiskette  
 abgespeichert.  
 \* PIP ist fertig

Legen Sie wieder die SoftCard Master Diskette ein und betaetigen:

**<RETURN>** Sie verlassen PIP  
**A>P:PATCHER** Starten Sie das Anpassungs-  
 programm

Dieses Programm liest das CP/M Betriebssystem aus der auf Ihrer Master Diskette vorhandenen Datei "CPM56.COM" ein, wandelt es so um, dass der gesamte AP 22 - Speicher genutzt wird, und schreibt es zusammen mit dem Ein-/Ausgabe System (BIOS) und einem System Generierungsprogramm in die Datei IBSCPM64.COM. Nach erfolgreicher Generierung meldet sich CP/M mit:

A>IBSCPM64 A:

IBSCPM64 schreibt das neu generierte CP/M System auf die System-Spuren der Diskette im Laufwerk A:

\*\*\* IBS CP/M SYSTEM GENERIERUNG \*\*\*  
\*\*\* V 1.3 - 15. 6. 1983 - R. E. \*\*\*  
\*\*\* auf IBS Apple ][ CP/M \*\*\*

Autostart Kommando oder <RETURN>

-><RETURN>

Disk in Laufwerk A: eingelegt (J/N)?

Quittieren Sie die Anfrage mit J, Ihre Master Diskette wird jetzt mit dem IBS Apple ][ CP/M System versehen.

IBS CP/M System auf Diskette geschrieben

Autostart Kommando oder <RETURN>

-><RETURN>

Disk in Laufwerk A: eingelegt (J/N)? N

A>

Jetzt haben Sie bereits eine eigenstartfaehige 64 K IBS Apple ][ CP/M Master Diskette. Zur vollen Ausnutzung der AP 22 Faehigkeiten muessen einige Programme noch angepasst werden; dazu sind die folgenden Angaben notwendig:

A>ERA P:PATCHER.COM Der Patcher wird geloescht  
A>PIP Starten Sie PIP

Legen Sie jetzt die "AP 22 - CP/M PATCHER" Diskette ein.

\*P:=\*.HEX[V] Die Anpassungsdateien werden in  
die interne Pseudodisk  
geschrieben.

\*P:=AEXEC.COM[OV] Das Autoexecute Programm wird  
in die interne Pseudodisk  
uebertragen.

\*P:=PATCH.AEX[V] Eine Kommandodatei zur weiteren  
Anpassung wird uebertragen.

\*  
Legen Sie jetzt wieder die Master Diskette ein

\*<RETURN> Verlassen Sie PIP

A>P: Schalten Sie auf die interne  
Pseudodisk um

P>AEXEC PATCH Durch dieses Kommando werden  
die verbleibenden Anpassungen  
durchgefuehrt.

Nachdem die Anpassungen ausgefuehrt worden sind, koennen Sie das  
64 K IBS Apple II System durch aus- und wieder einschalten des  
Rechners von Ihrer Master Diskette aus starten. SoftCard  
spezifische Programme die nicht angepasst worden sind, wie  
DOWNLOAD.COM, APDOS.COM, FORMAT.COM, COPY.COM sind unter AP 22  
CP/M nicht lauffaehig.

## 2.6 Der Disk Manager

Der Disk Manager ist das Formatier- und Kopierprogramm im AP 22 System. Der Formatierungsteil dient zum Erstellen neuer Apple II Disketten fuer das CP/M System. Vor ihrer ersten Benutzung muessen neue Disketten zunaechst ganz formatiert werden, es ist aber auch moeglich eine oder mehrere Spuren einer Apple II Disk oder Speicher Pseudodisk nachzuformatieren. Mit dem Kopierteil koennen exakte Kopien von Disketten erstellt werden. Dies wird hauptsaechlich zur Datensicherung verwendet.

Der Disk Manager kann in der CCP-Ebene mit

A>DM

gestartet werden. Er meldet sich daraufhin mit seinem Hauptmenue:

```
D I S K   M A N A G E R
Version 1.01 - 9.7.1983
```

Hauptmenue

```
F   Formatieren
K   Kopieren
E   Ende des Programms
```

Bitte waehlen Sie :

Sie koennen jetzt zwischen den 3 angegebenen Optionen waehlen:

1. Formatieren: mit dieser Option koennen Disketten, auch Pseudodisketten, ganz oder teilweise neu formatiert werden,
2. Kopieren: mit dieser Option koennen Kopien von Disketten erstellt werden und
3. Ende des Programms: mit dieser Option verlassen Sie den Disk Manager.

Durch Eingabe von CTRL-C (CTRL und C gleichzeitig druecken) koennen Sie bei jeder der folgenden Eingaben in das Hauptmenue zurueckkehren. Anstelle der angegebenen Gross- koennen auch Kleinbuchstaben verwendet werden. Die weiteren Eingaben der Formatier- und Kopier Optionen werden in den folgenden Abschnitten beschrieben.

2.6.1 Formatieren

Nach Angabe der Option F im Hauptmenue meldet sich der Disk Manager mit:

```

                Formatieren
    Diskette im Laufwerk
  
```

Geben Sie jetzt das Laufwerk, in dem die Diskette formatiert werden soll, an. Es koennen, neben Apple II Disk Laufwerken, auch die Pseudodisketten angewaehlt werden; diese brauchen Sie aber vor der ersten Benutzung nicht zu formatieren da dies vom AP 22 BIOS automatisch erfolgt. Bei unzuessaessigen Angaben (z. B. Laufwerk nicht vorhanden) wird erneut angefragt.

Haben Sie ein Laufwerk angegeben so erfolgt die Anfrage:

```

    G   ganz oder
    T   teilweise formatieren?
  
```

Bitte waehlen Sie :

Wenn Sie hier G angeben wird die Diskette im angegebenen Laufwerk ganz formatiert, bei T koennen einzelne Spuren angegeben werden.

Nur bei T erfolgt die Anfrage:

von Spur

Hier wird die erste Spur der Diskette, die formatiert werden soll, angegeben. Falsch eingegebene Ziffern koennen durch BS (ASCII 8, Taste <-) wieder geloescht werden. Die Eingabe der Ziffern wird durch ein beliebiges anderes Zeichen beendet (z. B. <RETURN>). Ist die angegebene Spur nicht vorhanden erfolgt eine neue Anfrage. Ist keine Ziffer angegeben worden, so wird als Voreinstellung Spur 0, bzw. bei wiederholter Benutzung die zuletzt angegebene, benutzt.

Anschliessend erfolgt die Anfrage (nicht bei Option G):

bis Spur

Hier wird die letzte Spur, die formatiert werden soll, angegeben. Als Voreinstellung wird hier die letzte, bzw. bei wiederholter Benutzung die zuletzt an dieser Stelle angegebene, Spur der Diskette angenommen. Ist die Endspur kleiner als die Anfangsspur erfolgt eine neue Anfrage, ansonsten gilt das gleiche wie bei "von Spur".

Jetzt kann mit dem Formatieren angefangen werden. Legen Sie jetzt Ihre Diskette im angegebenen Laufwerk ein und quittieren Sie die Anfrage

--Disk einlegen und <RETURN> druecken--

mit <RETURN>, waren die zuvor gemachten Angaben nicht korrekt, so koennen Sie auch hier durch CTRL-C ins Hauptmenue zurueckkehren!

Ist die eingelegte Diskette bereits formatiert - dies ist bei Pseudodisketten immer der Fall - so wird zusaetzlich angefragt:

\*\* Die Diskette wird geloescht \*\*  
Sind Sie sicher (J/N) ?

Wenn Sie hier J eingeben werden die Daten, die sich bereits auf der Diskette befinden, ueberschrieben und gehen damit verloren!

Bei teilweiser Formatierung werden natuerlich nur die angegebenen Spuren geloescht. Geben Sie J nur ein wenn Sie wirklich sicher sind!

Bei Eingabe von N wird erneut angefragt ob ganz oder teilweise formatiert werden soll.

Waehrend der Formatierung wird die gerade bearbeitete Spur auf der Console ausgegeben.

Nach erfolgreicher Formatierung erfolgt die Meldung:

--- Formatieren beendet ---

konnte die Diskette nicht formatiert werden:

\*\*\* Formatieren nicht moeglich \*\*\*

danach wird angefragt:

Noch eine Diskette formatieren (J/N)?

Wird hier J eingegeben so wird erneut angefragt ob ganz oder teilweise formatiert werden soll, bei N kehren Sie ins Hauptmenue zurueck; wurden die Optionen in der Kommandozeile angegeben (s. u.), so verlassen Sie den Disk Manager.

Neben den Eingaben bei der jeweiligen Anfrage, gibt es auch die Moeglichkeit diese beim Start des Disk Managers direkt in der Kommandozeile anzugeben. In diesem Fall werden die Optionen in der gleichen Reihenfolge hinter dem Programmnamen angegeben:

z. B.

A>DM FAG           Formatiere Diskette in A: ganz

A>DM FBT0,2       Formatiere Spur 0 bis 2 der Diskette in Laufwerk B:

A>DM FPG           Formatiere interne Pseudodisk ganz

## 2.6.2 Kopieren

Nach Angabe der Option K meldet sich der Disk Manager mit:

Kopieren

von Disk in Laufwerk

Geben Sie hier das Laufwerk an, in dem Ihre Quelldiskette eingelegt werden soll. Von diesem Laufwerk werden die Daten gelesen. Anschliessend wird angefragt:

auf Disk in Laufwerk

Geben Sie hier das Laufwerk, in dem Ihre Zieldiskette eingelegt werden soll. Die Daten werden von der Quelldiskette auf die Zieldiskette kopiert. Quell- und Ziellaufwerke duerfen identisch sein, in diesem Fall wird die jeweilige Diskette beim Kopieren angefordert, bitte schuetzen Sie Ihre Quelldiskette sicherheitshalber gegen Schreiben.

Kann vom angegebenen Quell- nicht auf das Ziellaufwerk kopiert werden, so meldet das Programm "\*\*\*FALSCHER ANGABE\*\*\*". Dies geschieht z.B. beim Versuch von der oder auf die interne Pseudodisk (Laufwerk P:?) zu kopieren.

Nach diesen Angaben wird das folgende Menue ausgegeben:

Was soll kopiert werden?

- S nur das System
- D nur Daten ohne System
- A alles oder
- T teilweise

Bitte waehlen Sie:

Je nach gewaehltm Quell- und Ziellaufwerk koennen einige dieser Optionen fehlen. Im einzelnen bewirken sie:

Option S: Die Systemspuren der Diskette werden kopiert. Voraussetzung hierfuer ist das beide Disketten ueber die gleiche Anzahl und Kapazitaet der Systemspuren verfuegen,

Option D: Der Datenbereich der Diskette wird kopiert. Voraussetzung ist das der Datenbereich der Zieldiskette groesser oder gleich dem der Quelldiskette ist. Es ist daher z. B. auch moeglich von einer 128 K Byte Pseudodiskette auf eine Apple ][ Disk oder von einer Apple ][ Disk auf eine 256 K Byte Pseudodiskette zu kopieren.

Option A: Es wird die gesamte Diskette kopiert. Voraussetzung dafür ist das Quell- und Ziellaufwerk den gleichen Typ haben, z. B. beide Apple II Disks.

Option T: Wie Option A, es kann aber zusätzlich angegeben werden welche Spuren kopiert werden soll(en). Wie die Spuren angegeben werden ist im Formatierungsteil unter Option T beschrieben.

Nach Eingabe der Option werden die Disketten angefordert. Sind Quell- und Ziellaufwerk unterschiedlich so erfolgt die Anfrage:

Disketten einlegen, dann <RETURN> bet.

Legen Sie jetzt Ihre Quell- und Zieldiskette in den angegebenen Laufwerken ein. Waren die vorherigen Angaben falsch, so koennen Sie an dieser Stelle durch Eingabe von CTRL-C wieder ins Hauptmenue zurueckkehren.

Waren Quell- und Ziellaufwerk gleich, so wird zunaechst die Quelldiskette mit der Anfrage:

QUELL Diskette einlegen, dann <RETURN>

angefordert. Versehen Sie bitte Ihre Quelldiskette mit dem Schreibschutz und legen Sie dann im angegebenen Laufwerk ein. Nach Eingabe von <RETURN> werden zunaechst Daten eingelesen und anschliessend die Zieldiskette mit der Meldung:

ZIEL Diskette einlegen, dann <RETURN>

Legen Sie jetzt Ihre formatierte Zieldiskette (ohne Schreibschutz) ein. Nach Betaetigen der <RETURN> Taste werden die zuvor gelesenen Daten auf diese Diskette geschrieben.

Dieser Vorgang wiederholt sich mehrmals. Nach Beendigung des Kopierens meldet sich der Disk Manager mit:

--- Kopieren beendet ---

Noch einmal kopieren (J/N) ?

Tritt waehrend des Kopierens ein Fehler auf, so wird mit:

\*\*\* Kopieren nicht moeglich \*\*\*

Noch einmal kopieren (J/N) ?

abgebrochen.

Wird die Anfrage mit J quittiert, so wird erneut die Kopieroption angefragt, bei N gelangen Sie wieder ins Hauptmenue es sei denn Sie haben die Optionen in der Kommandozeile angegeben, in diesem Fall verlassen Sie den Disk Manager.

Neben den Eingaben bei der jeweiligen Anfrage, gibt es auch die Moeglichkeit diese beim Start des Disk Managers direkt in der Kommandozeile anzugeben. In diesem Fall werden die Optionen in der gleichen Reihenfolge hinter dem Programmnamen angegeben:

z. B. (es wird angenommen das Laufwerk A: und B: Apple ][C Disks sind und Laufwerk C: eine 256 K Byte Pseudodisk ist)

A>DM KABA Diskette in A: auf Diskette in B: komplett kopieren

A>DM KBATD,2. Die Spuren 0 bis 2 der Diskette im Laufwerk B: werden auf die Spuren 0 bis 2 im Laufwerk A: kopiert

A>DM KBAS Diese Angabe bewirkt das gleiche wie die vorherige

A>DM KACD Kopiere Datenbereich der Diskette im Laufwerk A: auf die Pseudodisk im Laufwerk C:

## 2.7 Das Programm IBSCPM64.COM

Das Programm IBSCPM64.COM schreibt das IBS CP/M System auf die Boot Spuren einer Apple II Diskette.

### Aufruf:

IBSCPM64 <Laufwerk>:

<Laufwerk> steht fuer ein Apple II Disk Laufwerk, diese Angabe ist optional. Wird hier kein Laufwerk angegeben so wird von IBSCPM64 ein Laufwerk mit der Meldung:

Ziel Laufwerk oder <RETURN>

angefordert. Wird an dieser Stelle <RETURN> angegeben, so verlassen Sie IBSCPM64, andernfalls wird bei Angabe eines zulassigen Laufwerks normal fortgefahren.

Anschliessend wird ein Autostart Kommando mit der Meldung:

Autostart Kommando oder <RETURN>

->

angefordert. Hier kann ein normales CCP-Kommando angegeben werden. Dieses Kommando wird dann beim Kaltstart der Diskette automatisch ausgefuehrt. Wird <RETURN> angegeben, so wird kein Kommando eingetragen. Die maximale Kommandolaenge betraegt 63 Zeichen.

Nach dieser Eingabe erfolgt die Anfrage:

Disk in Laufwerk <Laufwerk>: eingelegt (J/N)?

Legen Sie jetzt die Diskette ein die mit dem System und dem evtl. angegebenen Autostart Kommando versehen werden soll. Nach der Eingabe J wird das System auf diese Diskette geschrieben. Durch Eingabe von N verlassen Sie das System wenn das Laufwerk in der Kommandozeile angegeben wurde, andernfalls wird ein anderes Laufwerk angefordert.

Jetzt muss die Meldung:

IBS CP/M System auf Diskette geschrieben

erscheinen. Ist dies nicht der Fall so wiederholen Sie das ganze bitte noch einmal. Anschliessend wird erneut ein Autostart Kommando angefordert, sofern das Laufwerk in der Kommandozeile angegeben worden ist; andernfalls wird erneut ein Laufwerk angefordert.

## 2.8 Das Selektiv-Loeschprogramm (ERAQ.COM)

Das Selektiv-Loeschprogramm ist eine Erweiterung des eingebauten ERA-Kommandos. Im Unterschied dazu wird bei ERAQ bei jeder Datei einzeln angefragt, ob geloescht werden soll.

### Aufruf:

ERAQ <name>.<ext>

Fuer <name>.<ext> kann jede zulaessige CP/M Dateibezeichnung incl. \* und ?-Zeichen stehen. Wird <name>.<ext> weggelassen so fragt ERAQ an, ob \*.\* verwendet werden soll. Wird die Anfrage nicht mit J quittiert, so wird das Programm beendet.

Fuer jede Datei, die mit der angegebenen Dateibezeichnung uebereinstimmt, wird einzeln angefragt, ob sie geloescht werden soll. Dies geschieht nur, wenn mit S quittiert wird. Zusaetzliche Dateiattribute, wie R/O und SYS, werden in der Anfrage angegeben, haben aber keinen Einfluss auf das Loeschen der Datei, d.h. auch schreibgeschuetzte Dateien (Attribut R/O) werden geloescht.

## 2.9 Das Autoexecute Programm (AEXEC.COM)

Das Autoexecute Programm ermöglicht es, mehrere Kommandos zusammenzufassen und dann automatisch auszuführen.

### Aufruf:

Form 1: AEXEC ufn parm#1 parm#2 ... parm#10

Form 2: AEXEC /

Form 3: AEXEC /kom#1;kom#2;...;

Form 4: AEXEC

### Form 1

Bei diesem Aufruf verhaelt sich AEXEC nahezu wie das SUBMIT-Programm (s. SoftCard Volume I S. 3-33 bis 3-35). Die Angabe ufn steht fuer eine Datei mit dem Typ ".AEX", die sich auf dem aktuellen Laufwerk befinden muss. In dieser Textdatei sind alle gewuenschten Kommandos aufgefuehrt und werden nach dem Aufruf von AEXEC sequentiell ausgefuehrt. Im Gegensatz zu SUBMIT werden alle Kommandos im Apple Speicher abgelegt und von dort anstelle der normalen Tastatur Eingabe geholt; es wird keine \$\$\$SUB-Datei angelegt.

In der AEX-Datei koennen "\$"- Parameter angegeben werden in der Form

\$1 \$2 \$3 ... \$10

Jede Stelle der AEX-Datei, die einen solchen Parameter enthaelt, wird beim Aufruf durch den in der Kommandozeile angegebenen Parameter "parm#u" (u gibt die Nummer an) ersetzt. \$-Zeichen koennen durch die Angabe \$\$ angegeben werden. Control-Zeichen koennen durch Voranstellen des Hochpfeils "^" und Angabe des Zeichens eingefuegt werden, z.B. ^A, ^C. Innerhalb einer AEX-Datei koennen wiederum andere AEXEC-Kommandos enthalten sein. Die dort angegebenen Kommandos werden dann eingefuegt und sofort abgearbeitet. Eine Schachtelung ist moeglich, solange der Speicher reicht. Ein Beispiel finden Sie in der SUBMIT-Beschreibung.

### Form 2

Mit diesem Aufruf gelangen Sie in den "interaktiven" Modus von AEXEC. Alle Kommandos koennen direkt von der Konsole eingegeben werden und werden nach einer Leereingabe automatisch ausgefuehrt. Innerhalb einer Zeile koennen Kommandos durch ; getrennt werden.

Beispiel:

```
A>AEXEC /<cr>
AUTOEXECUTE Version x.xx - dd.mm.yy
->DIR *.COM;STAT *.*<cr>
->MBASIC;10 FOR I=1 TO 1000<cr>
->20 NEXT;RUN;SYSTEM<cr>
-><cr>
```

Die angegebenen Kommandos werden jetzt nacheinander ausgeführt. Zunaechst wird DIR \*.COM, dann STAT \*.\* und anschliessend unter MBASIC eine Schleife von 1 bis 1000 ausgeführt.

Form 3

Diese Form entspricht der Form 2 mit dem Unterschied, dass nur 1 Kommandozeile angegeben werden kann.

Beispiel:

```
A>AEXEC /DIR *.HEX;STAT *.COM<cr>
```

Es werden die Kommandos DIR \*.HEX und STAT \*.COM nacheinander ausgeführt.

Form 4:

Bei diesem Aufruf wird eine kurze Zusammenfassung der Aufrufsmoeglichkeiten ausgegeben.

Zu Form 1 bis 3:

Nach Abarbeitung des AEXEC-Programms befindet sich der Rechner im Autoexecute Modus. Die Kommandos des Autoexecute Modus werden im Apple Speicher abgelegt. Der verwendete Bereich wird aber ebenfalls von der internen Pseudodisk (Laufwerk P:) belegt. Treten hier Konflikte auf, so wird bei Aufruf von AEXEC zunaechst angefragt, ob Laufwerk P: geloescht werden darf. Wird die Anfrage mit N quittiert, so wird AEXEC abgebrochen.

Im Autoexecute Modus werden bei der Konsolen-Eingabe die entsprechenden Zeichen aus dem Apple Speicher gelesen. Tastatureingaben werden jedoch ebenfalls, und zwar unmittelbar, beruecksichtigt. Dadurch ist es moeglich, die Konsolen-Ausgabe mit CTRL-S anzuhalten, andere Eingaben koennen jedoch die Kommandos veraendern! Durch Eingabe von CTRL-C kann der Autoexecute Modus jederzeit abgebrochen werden, ausgenommen, wenn dies nach Eingabe von CTRL-S erfolgt.

## 2.10 Das Programm UCOPY.COM

UCOPY dient zum Kopieren von Dateien in CP/M Systemen mit 1 Laufwerk.

### Aufruf:

UCOPY <name>.<ext> Q

Durch <name>.<ext> werden die zu kopierenden Dateien angegeben (? und \* sind zulaessig). Die Angabe Q ist optional, wird sie angegeben, so erfolgt vor dem Kopieren einer Datei zunaechst eine Anfrage. UCOPY laeuft nur auf Laufwerk A:. Nach dem Starten werden die Quell- und Zieldiskette jeweils nacheinander angefordert.

### Beispiel:

A>UCOPY \*.COM Q

Es werden alle .COM-Dateien der Quelldiskette auf die Zieldiskette kopiert, deren Anfrage mit S quittiert wird.

### 3. Das IBS CP/M System

#### 3.1 Starten des Systems

Das IBS CP/M System kann nach erfolgter Anpassung direkt durch Einschalten des Rechners gestartet werden. Hierzu muss eine Diskette mit dem IBS CP/M auf den Spuren 0 bis 2 in das Boot-Laufwerk (normalerweise Slot 6, Laufwerk 1) eingelegt werden. Anschliessend wird der Rechner eingeschaltet. Verfüegt Ihr Apple II ueber ein Autostart ROM, so ird das System automatisch geladen und gestartet, andernfalls muss das Monitor Kommando 6 <CTRL-P> eingegeben werden. Das System meldet sich mit der Startmeldung und A>.

Logfile	Startzeit	Systemzeit	Wartung
0000	00:00	00:00	
0001	00:00	00:00	
0002	00:00	00:00	
0003	00:00	00:00	
0004	00:00	00:00	
0005	00:00	00:00	
0006	00:00	00:00	
0007	00:00	00:00	
0008	00:00	00:00	
0009	00:00	00:00	
0010	00:00	00:00	
0011	00:00	00:00	
0012	00:00	00:00	
0013	00:00	00:00	
0014	00:00	00:00	
0015	00:00	00:00	
0016	00:00	00:00	
0017	00:00	00:00	
0018	00:00	00:00	
0019	00:00	00:00	
0020	00:00	00:00	
0021	00:00	00:00	
0022	00:00	00:00	
0023	00:00	00:00	
0024	00:00	00:00	
0025	00:00	00:00	
0026	00:00	00:00	
0027	00:00	00:00	
0028	00:00	00:00	
0029	00:00	00:00	
0030	00:00	00:00	
0031	00:00	00:00	
0032	00:00	00:00	
0033	00:00	00:00	
0034	00:00	00:00	
0035	00:00	00:00	
0036	00:00	00:00	
0037	00:00	00:00	
0038	00:00	00:00	
0039	00:00	00:00	
0040	00:00	00:00	
0041	00:00	00:00	
0042	00:00	00:00	
0043	00:00	00:00	
0044	00:00	00:00	
0045	00:00	00:00	
0046	00:00	00:00	
0047	00:00	00:00	
0048	00:00	00:00	
0049	00:00	00:00	
0050	00:00	00:00	
0051	00:00	00:00	
0052	00:00	00:00	
0053	00:00	00:00	
0054	00:00	00:00	
0055	00:00	00:00	
0056	00:00	00:00	
0057	00:00	00:00	
0058	00:00	00:00	
0059	00:00	00:00	
0060	00:00	00:00	
0061	00:00	00:00	
0062	00:00	00:00	
0063	00:00	00:00	
0064	00:00	00:00	
0065	00:00	00:00	
0066	00:00	00:00	
0067	00:00	00:00	
0068	00:00	00:00	
0069	00:00	00:00	
0070	00:00	00:00	
0071	00:00	00:00	
0072	00:00	00:00	
0073	00:00	00:00	
0074	00:00	00:00	
0075	00:00	00:00	
0076	00:00	00:00	
0077	00:00	00:00	
0078	00:00	00:00	
0079	00:00	00:00	
0080	00:00	00:00	
0081	00:00	00:00	
0082	00:00	00:00	
0083	00:00	00:00	
0084	00:00	00:00	
0085	00:00	00:00	
0086	00:00	00:00	
0087	00:00	00:00	
0088	00:00	00:00	
0089	00:00	00:00	
0090	00:00	00:00	
0091	00:00	00:00	
0092	00:00	00:00	
0093	00:00	00:00	
0094	00:00	00:00	
0095	00:00	00:00	
0096	00:00	00:00	
0097	00:00	00:00	
0098	00:00	00:00	
0099	00:00	00:00	
0100	00:00	00:00	

3.2 Das Ein-/Ausgabe System3.2.1 Logische und physikalische Gerate

Das CP/M Betriebssystem verfuegt ueber 4 Ein- bzw. Ausgabe-Geraete (CON:, LST:, RDR: und PUN:). Diese "Logischen" Gerate stehen fuer eines von mehreren physikalischen Geraten. Welches Gerat tatsaechlich angesprochen wird, bestimmt das sogenannte IOBYTE (Adr. 0003 im AP 22 Speicher). Im einzelnen sind folgende Zuordnungen moeglich (die Voreinstellungen sind unterstrichen).

Logisches Gerat	Funktion	physikalische Gerate	IOBYTE Nr.
CON:	Ein-/Ausgabe	TTY:	0
		CRT:	1
		BAT:	2
		UC1:	3
LST:	Ausgabe	TTY:	0
		CRT:	1
		LPT:	2
		UL1:	3
RDR:	Eingabe	TTY:	0
		PTR:	1
		UR1:	2
		UR2:	3
PUN:	Ausgabe	TTY:	0
		PTP:	1
		UP1:	2
		UP2:	3

Tabelle 1: Zuordnung logische -> physikalische Gerate

(\*) Beim Gerat BAT erfolgt die Eingabe ueber das logische Gerat RDR: und die Ausgabe ueber PUN:

Die Zuordnung der Gerate kann mit dem STAT Kommando erfolgen. In der Beschreibung des STAT Programms (s. 3 - 15 bis 3 - 22 Soft Card Volume I) finden Sie naehere Informationen.

Das IOBYTE selbst ist in 4 Felder aufgeteilt:

Bit 0 und 1 fuer die CON: IOBYTE Nummer,  
 Bit 2 und 3 fuer die RDR: IOBYTE Nummer,  
 Bit 4 und 5 fuer die PUN: IOBYTE Nummer und  
 Bit 6 und 7 fuer die LST: IOBYTE Nummer.

Durch die im IOBYTE angegebenen IOBYTE-Nummern aus Tabelle 1 werden die logischen den physikalischen Geraten zugeordnet.

### 3.2.2 Zuordnung der physikalischen Gerate

Die einzelnen Funktionen der physikalischen Gerate werden im AP 22 BIOS ueber eine Tabelle den verschiedenen Peripheriekarten in den Slots bzw. der eingebauten Tastatur und der 40 Zeichen Ausgabe zugeordnet. Nachfolgend soll die vorgegebene Standard-Einstellung beschrieben werden, (Informationen zur Aenderung der Zuordnungstabelle finden Sie in Kapitel 4.4). Welche Einzelfunktionen fuer ein Gerat vorhanden sind, haengt davon ab, welchem logischen Gerat es zugeordnet werden kann. Es gibt insgesamt 4 moegliche Einzelfunktionen:

1. Eingabe Status: Gibt an, ob ein Eingabe-Zeichen vorliegt
2. Eingabe eines Zeichens
3. Ausgabe Status: Gibt an, ob ein Zeichen ausgegeben werden kann
4. Ausgabe eines Zeichens

Verfuegt ein Gerat im AP 22 BIOS ueber die Funktion "Eingabe Status", so kann es nicht ueber die Funktion "Ausgabe Status" verfuegen. Dies stellt jedoch keine wesentliche Einschraenkung dar, da die "Ausgabe Status"-Funktion ohnehin nur selten benutzt wird.

Kann ein Gerat der Console (CON:) zugeordnet werden, so muessen die Funktionen "Eingabe Status Abfrage", "Eingabe eines Zeichens" und "Ausgabe eines Zeichens" vorhanden sein; kann es auf das List-Gerat (LIST:) gelegt werden, so muss die "Ausgabe eines Zeichens" und -optional- die Abfrage des "Ausgabe Status" eingebaut sein, bei Zuordnung zum Leser (RDR:) ist die "Eingabe eines Zeichens" und fuer den Schreiber (PUN:) die "Ausgabe eines Zeichens" erforderlich.

Die Standard Funktions Zuordnung ist in der nachfolgenden Tabelle angegeben:

Pysikalisches Geraet	Funktion	Zuordnung
TTY:	Eingabe Status	Apple Tastatur Status
	Zeichen Eingabe	Eingabe von Slot 3
	Ausgabe Status	immer nicht bereit
	Zeichen Ausgabe	Ausgabe auf Slot 3
CRT:	Eingabe Status	Apple Tastatur Status
	Zeichen Eingabe	Eingabe von Slot 3
	Ausgabe Status	immer nicht bereit
	Zeichen Ausgabe	Ausgabe auf Slot 3
UC1:	Eingabe Status	Apple Tastatur Status
	Zeichen Eingabe	Apple Tastatur Eingabe
	Zeichen Ausgabe	Ausgabe auf Slot 1
LPT:	Ausgabe Status	Status von Slot 1
	Zeichen Ausgabe	Ausgabe auf Slot 1
UC1:	Ausgabe Status	immer nicht bereit
	Zeichen Ausgabe	Ausgabe-Zeichen werden ignoriert
PTR:	Zeichen Eingabe	Eingabe von Slot 2
UR1:	Zeichen Eingabe	immer CTRL-Z
UR2:	Zeichen Eingabe	immer CTRL-Z
PTP:	Zeichen Ausgabe	Ausgabe auf Slot 2
UP1:	Zeichen Ausgabe	Ausgabe Zeichen werden ignoriert
UP2:	Zeichen Ausgabe	Ausgabe Zeichen werden ignoriert

Tabelle 2: Geraetefunktionen

Die Ein-/Ausgabe erfolgt ueber einen Slot, ist darin keine von BIOS erkannte Peripheriekarte installiert, so wird von der Apple Tastatur eingelesen und auf dem 40 Zeichen Video Interface ausgegeben. "Dummy" (= nur scheinbar vorhandene) Eingaben lesen immer CTRL-Z (ASCII SUB), die CP/M Datei Endekennung.

### Beispiele

Im Slot 3 ist eine 80 Zeichen Karte  
im Slot 1 ist ein paralleles Druckerinterface installiert

Wird die Console auf CRT: gelegt, so erfolgt die Eingabe ueber Slot 3 (Apple Tastatur) und die Ausgabe auf der 80 Zeichen Karte in Slot 3, Eingabe Status ist der Tastatur Status;  
wird die Console auf UC1: gelegt (Kommando: STAT CON:=UC1:), so erfolgt die Eingabe ueber die Apple Tastatur, die Ausgabe jedoch ueber das Druckerinterface in Slot 1, Eingabe Status ist wiederum der Tastatur Status.

### 3.2.3 Unterstuetzte Peripheriekarten

Vom AP 22 BIOS werden bevorzugt Peripheriekarten unterstuetzt, die ueber ein ROM im Bereich  $\$C \times 00 - \$C \times FF$  ( $x = \text{Slot-Nr.}$ ) verfuegen. Wird ein ROM erkannt, so vergleicht das BIOS in der BOOT-Phase maximal 3, in einer Konfigurationstabelle ausgegebene Werte ueberein, so wird in der spaeter beschriebenen Slottabelle eine entsprechende Kennung eingetragen. Im BIOS sind die Treiber fuer 4 derartig definierte Peripheriekarten vorhanden. Davon sind 3 Ein-/Ausgabekarten, die 4. ist der Apple ][ Disk Controller. Einen Ausdruck der 3 zugehoerigen Ein-/Ausgabetreiber finden Sie im Anhang A.

Es sind die folgenden Ein-/Ausgabetreiber vorhanden:

#### 1. fuer 80 Zeichen Karten, darunter fallen

IBS	AP 16-4	80 Zeichen Karte fuer Apple ][ plus
IBS	AP 2	Seriell Interface
IBS	AP 4G	Grafikfaehiges Druckerinterface
IBS	AP 11	Druckerinterface
IBS	AP 21	Grafikfaehiges Druckerinterface mit Spooler
IBS	AP 27	80 Zeichen Karte fuer Apple ][e
Videx	Videoterm	24 x 80 Video Terminal Card

und andere Peripheriekarten, die mit der gleichen Schnittstelle arbeiten. Die ROM-Kennung dieser Karten ist:

Byte  $\$C \times 05$ :  $\$38$  (alle Ausgaben als Hexadezimalzahl,  $x = \text{Slot-Nr.}$ )  
 Byte  $\$C \times 07$ :  $\$18$   
 Kennung in der Slottabelle: 4

Fuer diese Karten sind die Funktionen "Zeichen Eingabe" und "Zeichen Ausgabe" implementiert; als "Eingabe Status" und "Ausgabe Status" wird "nicht bereit" uebergeben.

#### 2. fuer serielle Interface Karten wie

Apple Communications Interface  
 California Computer Systems 7710 A Serial Interface  
 und andere Peripheriekarten mit gleicher Schnittstelle

Die ROM-Kennung dieser Karten ist:

Byte  $\$C \times 05$ :  $\$18$   
 Byte  $\$C \times 07$ :  $\$38$   
 Kennung in der Slottabelle: 3

Fuer diese Karten sind die Funktionen "Zeichen Eingabe", "Zeichen Ausgabe" und "Eingabe Status" implementiert; die Funktion "Ausgabe Status" liefert ebenfalls den Eingabe-Status und sollte daher nicht verwendet werden.

### 3. fuer parallele Drucker Interface Karten wie die

Apple Parallel Printer Card

und andere Peripheriekarten mit gleicher Schnittstelle.

Die ROM-Kennung dieser Karten ist:

Byte \$Cx05: \$48

Byte \$Cx07: \$48

Kennung in der Slottabelle: 5

Fuer diese Karten sind die Funktionen "Zeichen Ausgabe" und "Ausgabe Status" implementiert. Die Funktion "Zeichen Eingabe" liefert immer das Zeichen CTRL-Z (ASCII SUB = \$1A), die CP/M Dateiendekennung, die Funktion "Eingabe Status" liefert den Ausgabe Status und sollte daher nicht verwendet werden.

Die tatsaechlich aufgerufenen Funktionen dieser Peripheriekarten sind abhaengig von der Funktionszuordnung der physikalischen CP/M Gerate zu den Slots (s. Kapitel 3.2.2 und 4.4)

#### 3.2.4 Eingabe Umcodierung

Insbesondere zur Unterstuetzung der Apple Tastatur und des Microsoft Basic Interpreters ist ein Eingabe Umcodier Treiber eingebaut worden, der es ermoeglicht, einzelne Tastenfunktionen oder Tastenfolgen umzusetzen in andere Zeichen oder Zeichenfolgen. Dadurch ist es moeglich, die auf der Apple II Tastatur fehlenden Zeichen durch Betaetigen von 2 Tasten einzugeben oder aber auf die gleiche Weise laengere BASIC-Kommandos zu erhalten.

Im einzelnen werden (nur bei Eingabe ueber Slot 3 bzw. der Apple Tastatur) die folgenden Umcodierungen vorgenommen:

Eingebende Zeichen	Weitergereichte Zeichen
C	CTRL-K
<	CTRL-K
<ESC> A	C (Ae)
<ESC> a	c (ae)
<ESC> F	FILES
<ESC> f	FILES
<ESC> K	KILL"
<ESC> k	KILL"
<ESC> L	LOAD"
<ESC> l	LOAD"
<ESC> M	MBASIC
<ESC> m	MBASIC
<ESC> O	\ (Oe)
<ESC> o	(oe)
<ESC> R	RUN"
<ESC> r	RUN"
<ESC> S	SAVE"
<ESC> s	SAVE"
<ESC> U	] (Ue)
<ESC> u	) (ue)
<ESC> X	SYSTEM<CR>
<ESC> x	SYSTEM<CR>
<ESC> <BS>	<RUB>
<ESC> CTRL-O	CTRL-] (<GS>)
<ESC> -	_ (Underline)
<ESC> ?	CTRL-_ (<US>)
<ESC> Q	CTRL-A
<ESC> q	CTRL-A
<ESC> <ESC>	<ESC>

Tabelle 3: Standard Eingabe Umcodierungen

Erläuterungen:

<..> stellt nicht die Zeichenfolge <..>, sondern die Bezeichnung fuer ein ASCII-Steuerzeichen dar, dies gilt ebenso fuer die Angabe CTRL-... Eingeklammerte Werte sind andere Bezeichnungen fuer das gleiche Zeichen.

Die hier angegebenen Umcodierungen koennen vom Benutzer veraendert werden. Zu diesem Zweck ist der Quellcode der Umcodier-Tabelle beigefuegt worden (s. Anhang B).

Der Aufbau dieser Tabelle soll im folgenden erlaeutert werden:

Der Eingabe Umcodier Treiber benutzt zunaechst die "ADRESSTABELLE FUER EINGABE UMCODIERUNGEN" um zu ermitteln, ob ueberhaupt eine Umcodierung vorgenommen werden soll. Diese Tabelle besteht aus 8 Eintraegen zu je 16 Bit. Jeder dieser Eintraege steht fuer Eingaben von einem Slot; der erste Eintrag fuer die Apple Tastatur, der 2. fuer Eingaben von Slot 1, der 3. fuer Eingaben von Slot 2, ..., der 8. fuer Eingaben von Slot 7. Ist der jeweilige Eintrag 0, so wird keine Umcodierung vorgenommen, andernfalls gibt er die Adresse der Umcodier-Tabelle im Apple Speicher an.

Die eigentliche Umcodier-Tabelle gliedert sich in 3 Teile:

Die Einleitungstabelle, sie enthaelt,

1. Definitionen der vorzutippenden Zeichen (z.B. in der Standard-Konfiguration <ESC>)
2. Die Erkennungstabelle, sie beinhaltet die umzucodierenden Zeichen (bei laengeren Tastenfolgen nur das letzte Zeichen) und einen Verweis auf die Weitergabe-Tabelle
3. Die Weitergabe-Tabelle, sie beinhaltet die Zeichen, die anstelle der erkannten Tastenfolge als eingegebene Zeichen weitergegeben werden.

Die 3 Teile bilden eine Einheit und muessen in dieser Reihenfolge definiert werden.

Die Einleitungstabelle besteht aus Eintraegen mit einer Laenge von 3 Byte; sie endet mit 1 Byte mit Inhalt \$FF. Das erste der 3 Byte gibt das vorher einzugebende Zeichen an, die naechsten 2 Byte die Adresse einer anderen Eingabe Umcodiertabelle, die verwendet werden soll, wenn dieses Zeichen eingegeben wird. Ist dies der Fall, so wird die dort angegebene Tabelle benutzt, die wiederum mit einer Einleitungstabelle, in der wieder ein weiteres einzugebendes Zeichen und eine andere Tabelle definiert sein koennen, beginnt. Auf diese Weise ist es moeglich, auch laengere Eingabesequenzen zu erkennen. Das jeweils letzte Zeichen wird in der Erkennungstabelle festgelegt. Eine leere Einleitungstabelle besteht aus 1 Byte \$FF.

In der Erkennungstabelle wird im 1. Byte die Anzahl der angegebenen Zeichen definiert. Anschliessend werden alle Zeichen, die umcodiert werden sollen, nacheinander ASCII-codiert aufgefuehrt. Das Bit 7 des ASCII-Codes ist normalerweise unbenutzt, es gibt hier, wenn es gesetzt ist, an, dass das angegebene Zeichen in Gross- und Kleinschreibung verwendet werden soll. Hinter den Zeichenangaben werden fuer jedes Zeichen die relativen Abstaende des weiterzugebenden Textes zum Anfang der Weitergabe-Tabelle definiert. Der 1. hier angegebene Wert ist der Abstand fuer das 1. Zeichen usw. Das Ende dieser Tabelle ist der Anfang der Weitergabe-Tabelle.

In der Weitergabe-Tabelle werden die weiterzugebenden Zeichen in umgekehrter Reihenfolge definiert, d. h. letztes Zeichen zuerst, erstes Zeichen zuletzt. Die Gesamlaenge der Tabelle darf 255 Byte nicht ueberschreiten. Das erste Byte jedes Eintrages gibt die Laenge des folgenden Textes in Byte an, die Anfangsposition ergibt sich aus den Eintraegen in der Erkennungstabelle.

Als Beispiel (zur weiteren Veranschaulichung) soll die im Anhang B zu findende Umcodierungstabelle dienen.

### 3.2.5 Ausgabe Umcodierung

Die meisten Video Terminals, darunter auch der Apple ][ 24 x 40 Zeichen Bildschirm, unterstützen Sonderfunktion wie direkte Cursor-Adressierung, Inverse Darstellung, Bildschirm Loeschen usw. Diese Funktionen koennen im AP 22 CP/M System auf einfache Weise durch Benutzerprogramme genutzt werden.

Bildschirmfunktionen werden durch Ausgabe von einem oder mehreren Steuerzeichen aufgerufen. Diese Steuerzeichen sind jedoch leider nicht festgelegt und werden somit von jedem Hersteller anders implementiert. Insbesondere die populaeren 80-Zeichen Karten benoetigen Zeichen, die von nahezu keinem kommerziellen CP/M-Programm unterstuetzt werden. Um die Bildschirmfunktionen trotzdem nutzen zu koennen, wurde die Ausgabe Umcodierung eingebaut. Diese bewirkt die Umsetzung der vom Programm ausgegebenen Steuerzeichen in Steuerzeichen, die fuer das Terminal bzw. die 80 Zeichen Karte notwendig sind. In der Standard-Konfiguration werden die Steuerzeichen fuer ein SOROC IQ 120 Terminal vom Ausgabe Umcodierungs-System erkannt und in die Steuerzeichen fuer eine Videx Videoterm 24 x 80 Zeichen Karte umgesetzt, d. h. die Bildschirm Funktionen dieser Karte koennen ueber Steuerzeichen fuer das weitverbreitete SOROC IQ 120 Terminal angesprochen werden. Dieses Terminal wird von nahezu allen kommerziellen bildschirmorientiert arbeitenden Programmen unterstuetzt.

Insgesamt werden 9 Bildschirmfunktionen unterstuetzt:

1. Bildschirm Loeschen
2. Loeschen bis zum Seitenende
3. Loeschen bis zum Zeilenende
4. Normaldarstellung (Text weiss auf schwarz)
5. Inverse Darstellung (Text schwarz auf weiss)
6. Cursor zum Seitenanfang (links oben) bewegen
7. Absolute Cursorpositionierung
8. Cursor um 1 Zeile nach oben bewegen
9. Cursor um 1 Zeichen nach rechts bewegen (Zeichen wird nicht geloescht)

Es wird angenommen, dass das Zeichen Backspace (ASCII 8) den Cursor um 1 Stelle nach links, und das Zeichen Zeilenvorschub (ASCII 10) um 1 Zeile nach unten bewegt.

Diese Funktionen werden in der Standard Konfiguration bei Ausgaben auf dem 24 x 40 Zeichen Bildschirm oder auf Slot 3 wie folgt umgesetzt:

Funktion	Erkannte Steuerzeichen (SOROC IQ120)	Weitergereichte Steuerzeichen (Videx 24 x 80)
1	<ESC> *	<FF> (CTRL-L)
2	<ESC> Y	<VT> (CTRL-K)
3	<ESC> T	<GS> (CTRL-])
4	<ESC> )	<SO> (CTRL-N)
5	<ESC> (	<SI> (CTRL-O)
6	<RS> (CTRL-^)	<EM> (CTRL-Y)
7	<ESC> =	<RS> (CTRL-^)
8	<VT> (CTRL-K)	<US> (CTRL-_)
9	<FF> (CTRL-L)	<FS> (CTRL-\)

Tabelle 4: Standard Ausgabe Umcodierungen

Erläuterungen:

<..> stellt nicht die Zeichenfolge <..> sondern die Bezeichnung eines ASCII-Steuerzeichen dar, dies gilt ebenso fuer die Angabe CTRL-... Eingeklammerte Werte sind andere Bezeichnungen fuer das gleiche Zeichen.

Zur Umcodierung der Ausgabe Zeichen verwendet das AP 22 BIOS eine Umcodiertabelle. Diese teilt sich in 2 gleich aufgebaute Teile:

1. Die "Software" Funktionstabelle: sie enthaelt die Steuerzeichen, die vom BIOS erkannt werden, und
2. Die "Hardware" Funktionstabelle: sie enthaelt die Steuerzeichen, die anstelle der, in der Software Tabelle angegebenen, tatsaechlich ausgegeben werden.

Als Steuerzeichen kann entweder 1 Control-Zeichen (0-31) oder ein Control-Zeichen gefolgt von einem beliebigen ASCII-Zeichen angegeben werden. Die Umcodiertabelle fuer die Standard Konfiguration finden Sie in Anhang B, die im folgenden Abschnitt fett angegebenen Worte sind Label aus dieser Tabelle.

Die Umcodiertabelle beginnt mit dem Label UMOTB. In den Spalten Hardware und Software ist der Abstand in Byte von diesem Label angegeben:

Control-Zeichen	Hardware	Software
0	0	0
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20
21	21	21
22	22	22
23	23	23
24	24	24
25	25	25
26	26	26
27	27	27
28	28	28
29	29	29
30	30	30
31	31	31

Funktion	Software	Hardware	Beschreibung
zu 7	0	11	Dieser Wert wird bei absoluter Cursorpositionierung zu den X und Y Koordinaten addiert. Er darf zwischen 0 und 127 liegen, gebräuchlich sind 0 und 32. Bit 7 gibt an, in welcher Reihenfolge die x/y Koordinaten ausgegeben werden. Ist Bit 7 = 0, so wird zuerst die y- und dann die X-Koordinate ausgegeben; ist Bit 7 = 1, so wird erst der X- und dann der Y-Wert ausgegeben.
	1	12	Einführungszeichen. Bei Steuersequenzen, die aus mehr als einem Zeichen bestehen, wird dieses Zeichen als erstes erwartet.
<p><b>Hinweis:</b> In den nachfolgenden Funktions-Steuerzeichen Einträgen wird zuerst das Einführungszeichen erwartet, wenn Bit 7 = 1 ist; ist der Eintrag = 0, so ist die Funktion nicht vorhanden und wird vom AP 22 BIOS ignoriert.</p>			
1	2	13	Bildschirm loeschen
2	3	14	Loeschen bis zum Seitende
3	4	15	Loeschen bis zum Zeilende
4	5	16	Normaldarstellung
5	6	17	Inverse Darstellung
6	7	18	Cursor nach links oben (HOME)
7	8	19	Absolute Cursorpositionierung (s.o.)
8	9	20	Cursor 1 Zeile hoch
9	10	21	Cursor nach rechts ohne Loeschen

Tabelle 5: Bildschirm Funktionen

Fuer den Apple 24 x 40 Zeichen Bildschirm sind alle 9 Funktionen implementiert, die Hardware Zeichen werden hierfuer nicht benutzt. Wird als Software Zeichen eine 0 angegeben, so ist die Funktion auch auf dem 24 x 40 Zeichen Bildschirm nicht mehr verfuegbar; Angaben fuer Hardware Zeichen sind fuer den Apple Bildschirm ohne Bedeutung und koennen in diesem Fall fuer Peripheriekarten (z.B. 80 Zeichen Karte) definiert werden.

Grundsätzlich kann fuer jeden Slot eine Ausgabe Umcodiertabelle definiert werden, sinnvoll ist dies natuerlich nur dann, wenn in diesem Slot auch tatsaechlich eine Ausgabe Peripheriekarte steckt und wenn fuer diese ein CP/M Geraet definiert ist.

Die Slot-Tabellenzuordnung wird in der "ADRESSTABELLE FUER AUSGABE UMCODIERUNG" angegeben. Diese besteht aus 8 Eintraegen zu je 16 Bit. Der 1. Eintrag gibt die Adresse einer Umcodiertabelle fuer die 24 x 40 Zeichen Ausgabe an, der 2. fuer Ausgabe auf einer Peripheriekarte im Slot 1 (z.B. Drucker), der 3. fuer Slot 2, usw. Ein Eintrag besteht aus der Adresse der Ausgabe Umcodiertabelle im Apple Speicher oder aus 0. Im 2. Fall wird nichts umcodiert.

### 3.2.5.1 Terminal unabhaengige Bildschirm Funktions-Ausgabe und Cursor-Adressierung

Da im AP 22 BIOS die Steuerzeichen fuer das tatsaechlich vorhandene Terminal in der Ausgabe Umcodiertabelle angegeben werden, ist es durch Verwendung dieser Tabelle moeglich, eine von den Steuerzeichen unabhaengige Funktionsausgabe zu realisieren. Das folgende Programm stellt ein Assembler Unterprogramm dar, mit dessen Hilfe Bildschirmfunktionen auf Slot 3 ausgegeben werden koennen, ohne die Steuerzeichen zu kennen. Das Unterprogramm SCRFUN erwartet im Register E die Bildschirm Funktions Nr. wie in Tabelle 5 angegeben, GOTOXY positioniert den Cursor an die Stelle (Inhalt Register H = X-Koordinate, Inhalt Register L = Y-Koordinate). Die obere linke Ecke des Bildschirms hat die Koordinaten (0,0).

```

;
;*****
;
;   Unterprogramm zur Terminal unabhaengigen
;   Bildschirm-Funktionsausgabe und Cursor Positionierung
;   (Es werden die Steuerzeichen fuer 24x40 Zeichen verwendet)
;
;   (R. E. - 28. 5. 83)
;
;*****
;
;   Konstanten
;
0005 =      BDOS   EQU      0005H      ;CP/M Funktions Aufruf
0C18 =      OADR7B EQU      0C18H      ;Ausgabe Tabellen Adresstabelle
0000 =      CUROFFS EQU      0          ;Distanz zum Cursor Offset
0001 =      LEADIN  EQU      1          ;Distanz zum einfuehrenden Zeichen
0002 =      FUNTAB  EQU      2          ;Distanz zur Funktions Tab.

```

-----  
 SCRIFUN

Mit dem Unterprogramm SCRIFUN koennen Bildschirm-  
 Funktionen Terminal unabhangig ausgegeben werden.

Register E enthaelt beim Aufruf die Funktionsnummer (1-9)

Das Unterprogramm versendet alle Erstregister des Z80 !!

-----

```

0000 07 SCRIFUN: RST      2           ;Tabellenanfang holen
0001 190C          DN      0AD7B     ;LSB der Adresse
0003 4F          MOV     L,A         ;in L
0004 07          RST      2           ;
0005 190C          DN      0AD7B+1   ;MSB der Adresse
0007 57          MOV     H,A         ;in H
0008 E5          PUSH    H           ;Tabellenanfang retten

0009 1588          MUI     D,8           ;DE = Funktionsnummer
000B 19          DAD     0           ;+ Tabellenanfang
000C 23          INC     H           ;+ 1 = Funktionseintrags-Adresse

000D E7          RST      4           ;Funktionseintrag holen
000E 82          DB      2           ;
000F E1          POP     H           ;Tabellenanfang wieder in HL
0010 84          CMP     0           ;= 8 (nicht definiert) ?
0011 C8          RZ                ;Ja => *

0012 388A          DB      38H, COUTA-6-1 ;kein Einfuehrungszeichen -)

;
; Vor Ausgabe des Funktionszeichens muss das Einfuehrungszeichen
; ausgegeben werden.
;
0014 E47F          ANI     7FH
0015 F5          PUSH    PSW           ;Funktionszeichen retten
0017 23          INC     H
0018 E7          RST      4           ;Einfuehrungszeichen holen
0019 82          DB      2           ;
001A CD1E88       CALL    COUTA         ;Einfuehrungszeichen ausgeben
001D F1          POP     PSW           ;Funktionszeichen zurueckholen

;
001E 5F          COUTA: MOV     E,A         ;Zeichen fuer 8005 Aufruf umspeichern
001F E5          COUTE: PUSH    H         ;HL Wert retten
0020 8E82          MUI     C,2           ;CONSOLE Output Funktion
0022 CD8508       CALL    8005         ;
0025 E1          POP     H           ;HL zurueckholen
0026 C9          RET

```

```

:
: *****
:
: GOTOXY
:
: Mit Hilfe dieses Unterprogramms kann der Cursor Terminal-
: unabhangig absolut positioniert werden.
:
: Bei Aufruf des Unterprogramms enthaelt:
: Register H : X-Koordinate
: Register L : Y-Koordinate
:
: Das Unterprogramm veraendert alle Erstregister des Z80 !!
:
: -----

```

```

0027 E5 GOTOXY: PUSH H ;Koordinaten retten
0028 1E07 MOV! E,7
002A CD0000 CALL SCRFUN ;Absolute Positionierung einleiten
:
: HL enthaelt die Anfangsadresse der Ausgabeumcodier Tabelle + 1
:
002D 2B DCX H ;auf Offset Adresse
002E E7 RST 4 ;Offset holen
002F 02 DB 2
0030 E1 PDP H ;Koordinaten zurueckholen
0031 07 ORA A ;Reihenfolge aendern ?
0032 F20000 JP NORVS ;Nein ->
:
: Reihenfolge der Koordinaten umkehren
:
0035 5D MOV E,L
0036 6C MOV L,H
0037 63 MOV H,E
:
0038 5F NORVS: MOV E,A ;Offset in E retten
0039 04 ADD H ;A = 2. Koordinate + Offset
003A 67 MOV H,A ;fuere spaeter retten
003B 7B MOV A,E
003C 05 ADD L ;A = 1. Koordinate + Offset
003D CD1E00 CALL COUTA ;1. Koordinate ausgeben
0040 5C MOV E,H
0041 C31F00 JHP COUTE ;2. Koordinate ausgeben

```

Anstelle des im Programm 1 angegebenen Unterprogramms SCRFUN kann auch eine speziell implementierte Funktion benutzt werden. In diesem Fall wird die Funktion auf dem Gerat ausgeführt, auf dem auch die letzte Ausgabe erfolgte, d.h. vor Benutzung dieser Funktion muss mindestens 1 Zeichen auf dem gewünschten Gerat ausgegeben worden sein.

```

;*****
;
; Unterprogramm zur Terminal unabhängigen
; Bildschirm-Funktionsausgabe
; (Es werden die Steuerzeichen fuer den zuletzt
; benutzten Slot verwendet)
;
; (R. E. - 28. 5. 83)
;*****
;
; Konstanten
;
0005 = 8005 EQU 0005H ;CP/M Funktions Aufruf
0049 = 0049 EQU 0049H ;288 Seite 8 Aufruf
;
0C18 = 0C18 EQU 0C18H ;6502 Adresse
;*****
;
; SCRFUN
;
; Mit dem Unterprogramm SCRFUN koennen Bildschirm-
; Funktionen Terminal unabhangig ausgegeben werden.
;
; Register E enthaelt beim Aufruf die Funktionsnummer (1-9)
;
; Das Unterprogramm veraendert alle Erstregister des 288 !!
;-----
0000 C34900 SCRFUN: JMP 0049H ;BIOS-Unterprogramm aufrufen

```

```

:-----:
:
: GOTDXY
:
: Mit Hilfe dieses Unterprogramms kann der Cursor Terminal-
: unabhängig absolut positioniert werden.
:
: Bei Aufruf des Unterprogramms enthaelt:
: Register H : X-Koordinate
: Register L : Y-Koordinate
:
: Es wird der Offset der 24x48 Umcodier Tabelle verwendet
:
: Das Unterprogramm veraendert alle Erstregister des Z80 !!
:-----:
:
0003 E5 GOTDXY: PUSH H ;Koordinaten retten
0004 E6 MVI E,7
0005 C00000 CALL SCRFUN ;Absolute Positionierung einleiten
0006 07 RST 2 ;Tabellenanfang holen
0007 180C 0M 0ADRTB ;LSB der Adresse
0008 6F MOV L,A ;in L
0009 07 RST 2
000A 190C 0M 0ADRTB+1 ;MSB der Adresse
000B 67 MOV H,A ;in H
:
: HL enthaelt die Anfangsadresse der Ausgabeumcodier Tabelle
:
0011 E7 RST 4
0012 42 0B 2 ;Offset holen
0013 E1 POP H ;Koordinaten zurueckholen
0014 B7 ORA A ;Reihenfolge sendern ?
0015 F21000 JP NORUS ;Nein ->
:
: Reihenfolge der Koordinaten umkehren
:
0018 50 MOV E,L
0019 6C MOV L,H
001A 43 MOV H,E
:
001B 5F NORUS: MOV E,A ;Offset in E retten
001C 34 ADD H ;A = 2. Koordinate + Offset
001D 47 MOV H,A ;fuer spaeter retten
001E 78 MOV A,E
001F 35 ADD L ;A = 1. Koordinate + Offset
0020 5F MOV E,A
0021 C02500 CALL COUTE ;1. Koordinate ausgeben
0022 5C MOV E,H ;2. Koordinate ausgeben
:
:
0025 E5 COUTE: PUSH H ;HL Wert retten
0026 4E02 MVI C,2 ;Console Output Funktion
0027 C00500 CALL BDOS
0028 E1 POP H ;HL zurueckholen
0029 C9 RET

```

### 3.3 Disk Ein- / Ausgabe

In welchen Slots Disk Controller installiert werden koennen und als welches CP/M Laufwerk sie dann im System erscheinen, wurde bereits in den Abschnitten 2.1 und 2.2 beschrieben. Hier soll zunaechst die grundsaeztliche Arbeitsweise des AP 22 BIOS bei Disk Ein- / Ausgabe dokumentiert werden.

Das CP/M Betriebssystem teilt die Disketten in Sektoren zu je 128 Byte Speicherkapazitaet ein. Mit dieser Sektorgroesse kann in allen CP/M Benutzerprogrammen gearbeitet werden, fuer die Disk Controller Treiberprogramme ist dies allerdings zu klein. Die Apple II Disk wird grundsaeztlich mit 256 Byte grossen Sektoren formatiert. Aus diesem Grund ist im AP 22 BIOS ein Programm enthalten, das je 2 128 Byte Sektoren zu einem 256 Byte Sektor zusammenfasst. Dadurch bedingt muessen Disk Treiber im AP 22 BIOS grundsaeztlich immer mit 256 Byte/Sektor arbeiten. Insgesamt unterstuetzt das BIOS die Funktionen Selektierung des Laufwerks, Deselektierung des Laufwerks, Positionieren auf eine Spur, Lesen eines 256 Byte Sektors, Schreiben in einen 256 Byte Sektor und optional Formatieren einer Spur. Bei den letzten 3 Funktionen und beim Selektieren werden Fehlermeldungen des Treibers durch das BIOS Fehlermeldungssystem unterstuetzt.

Der physikalische Aufbau des Disk Controllers wird durch den Disk Parameter Block (DPB) festgelegt. Dieser Block wird sowohl vom BIOS wie auch vom AP 22 BIOS zur Ermittlung der Spur- und Sektoranzahl benutzt. Wie dieser im einzelnen aufgebaut ist, finden sie im "CP/M Alteration Guide", einem Teil der CP/M Dokumentation. Die genaue Spezifikation der Disk Treiber Schnittstelle finden Sie im Abschnitt 4.5.

#### 3.3.1 Zuordnung der Disk Laufwerke

Wie kann die Zuordnung der Disketten Laufwerke zu den einzelnen Slots geaendert werden?

Zu diesem Zweck befindet sich im AP 22 BIOS eine entsprechende Konfigurationstabelle. In dieser Tabelle kann angegeben werden welches CP/M Laufwerk (A: bis D:) welchen Slot, und in diesem Slot, welchem Laufwerk zugeordnet wird. Den Ausdruck der Disk Konfigurations Tabelle mit der Standard Konfiguration finden Sie im Anhang D.

Wie ist diese Tabelle aufgebaut?

Sie besteht aus Eintraegen zu je 3 Byte fuer jedes der 4 unterstuetzten CP/M Laufwerke. Dabei gibt an:

**Byte 0:** Ein Wert zwischen 40H und 52H falls das Laufwerk vorhanden ist, andernfalls OFFH. Dieser Wert wird vom BIOS in der Kaltstart Phase eingesetzt, somit ist der Initialwert in der Tabelle irrelevant.

**Byte 1:** Dies ist der eigentlich konfigurierbare Wert in dieser Tabelle. Der Wert besteht aus der Slot-Nr. \* 16 + Laufwerk Nummer, wobei Laufwerk 0 das 1., Laufwerk 1 das 2., usw. bis Laufwerk 15 das 16. angeschlossene Laufwerk angibt und die Slot-Nr.

zwischen 1 und 7 liegen kann.

**Byte 2:** Dieser Wert gibt an, wie oft ein Lese- oder Schreibkommando wiederholt wird, bevor ein Fehler gemeldet wird. An dieser Stelle wird der Wert vom BDOS in der Kaltstart Phase eingesetzt, somit ist der Initialwert in der Tabelle irrelevant.

**Byte 3,4:** Diese 2 Byte bilden einen 16 Bit Wert (Byte 3: LSB, Byte 4: MSB). Er gibt an, welche Spur des Laufwerks als letzte angesprochen wurde. -1 (OFFFHH) gibt an, dass das Laufwerk bislang nicht benutzt wurde. Der Initialwert muss OFFFHH sein.

In dieser Tabelle ist somit also nur Byte 1 konfigurierbar, die anderen Werte werden zum Teil in anderen Tabellen angegeben. Der 1. Eintrag steht fuer CP/M Laufwerk A:, der 2. fuer B:, der 3. fuer C:, der 4. fuer D:, es ist auch ein 5. Eintrag vorhanden der fuer die interne Pseudodisk, Laufwerk P:, steht, in diesem Fall muss Byte 1 eine 0 enthalten. Die -1 in den Bytes 3 und 4 wird uebrigens zur Rekalibrierung der Laufwerke bzw. zur Initialisierung der Pseudodisk benutzt und ist daher unbedingt notwendig. Die Tabelle mit der Standard Konfiguration ist als Datei DSKZUORD.ASM auf der AP 22 - CP/M PATCHER Diskette enthalten. Nach dem Kaltstart finden Sie die Disk Zuordnungstabelle ab Adresse OFF18H im AP 22 Speicher, sie sollte aber nicht dort geaedert werden. Wird die Datei IBSCPM64.COM mit DDT, ohne Angabe eines Offsets, in den AP 22 Speicher geladen, so faengt sie in Adresse OB18H an.

### 3.3.2 Unterstuetzte Disk Controller

Im einzelnen werden die folgenden Disk Controller in der derzeitigen BIOS-Version unterstuetzt:

#### a) Apple JC Disk Controller mit 1 oder 2 Laufwerken

Dieser Controller wird wie im SoftCard CP/M unterstuetzt. Die Formatierung der Disketten entspricht der im Apple DOS 3.3, im SoftCard CP/M mit 16 Sektoren und im Apple Pascal 1.1 verwendeten. Die insgesamt 35 Spuren der Laufwerke werden in 3 Systemspuren (Spur 0 bis 2) und 32 Datenspuren (Spur 3 bis 34) aufgeteilt. Jede Spur enthaelt 16 Sektoren zu je 256 Byte, das entspricht 32 CP/M Sektoren zu je 128 Byte. Die 256 Byte Sektoren werden mit dem Skew-Faktor 3 angesprochen, d. h. nach Sektor 0 Sektor 3 dann Sektor 6 usw. Die Datenkapazitaet ohne Systemspuren betraegt pro Laufwerk 128 K Byte.

#### b) IBS Speicher Pseudodisk

Als IBS Speicher Pseudodisk koennen mehrere IBS Karten verwendet werden:

AP 10	64 K Byte Speicher mit 68B09 CPU
AP 13a,b	64 oder 128 K Byte Speicherkarte
AP 17	64, 128, 196 oder 256 K Byte Speicherkarte
AP 20	128 K Byte Speicher mit 68000 CPU
AP 21	64 K Byte Speicher mit 6511 CPU (s. u.)
AP 25	128 K Byte Speicher mit 8086/87 CPU

Wenn eine dieser Karten in einem Slot installiert wird dem 1 oder 2 CP/M Laufwerke zugeordnet sind (s. Abschnitt 3.3.1), so kann sie, ohne zusaetzliche Installationsprogramme, sofort als Speicher Pseudodiskette verwendet werden. Ist dem Slot 1 Laufwerk zugeordnet, so steht das zugehoerige CP/M Laufwerk mit der gesamten Speicherkapazitaet der Karte zur Verfuegung, sind dem Slot 2 Laufwerke zugeordnet, so wird der Speicher zu gleichen Teilen auf jedes Laufwerk verteilt (je Laufwerk 50% des Speichers). Der Einsatz von mehr als 1 IBS Speicherkarte ist nur mit Einschränkungen moeglich. In diesem Fall muessen alle CP/M Laufwerke die einer Speicherkarte zugeordnet sind die gleiche Kapazitaet haben. Der Stromverbrauch der Peripheriekarten macht es in diesem Fall erforderlich ein staerkeres Netzteil einzusetzen mit einer Leistung von min. 5 A auf der 5 V Versorgungsspannung. Die Laufwerke werden in Spuren zu je 1 K Byte mit je 4 Sektoren zu 256 Byte eingeteilt, es gibt keine Systemspuren.

Anmerkungen zur AP 21:

Die AP 21 kann nur dann als 64 K Pseudodiskette eingesetzt werden, wenn sie nicht vom BIOS als Druckerspooiler erkannt wird. Dazu ist es erforderlich das Spooler EPROM (ROM Bereich 5CxNN) zu entfernen, die Fassung muss leer sein.

Anmerkung zur AP 10 und AP 21 auf BASIS 108:

Der Einsatz dieser Karten als 64 K Pseudodisk ist auf diesem Rechner nur moeglich, wenn diese Karten in einem Slot mit hoeherer Nummer als die AP 22 installiert sind. Andernfalls werden diese Karten als AP 22 erkannt, das System ist in diesem Fall nicht mehr startbar.

### 3.3.3 Die eingebaute Pseudodiskette

Im AP 22 BIOS steht der nicht belegte Apple ][ Speicher fuer den CP/M Benutzer als Pseudodiskette zur Verfuegung. Diese kann, nach dem Kaltstart, sofort, ohne ein zusaetzliches Installationsprogramm, benutzt werden. Die Pseudodiskette wird als CP/M Laufwerk P: angesprochen und hat eine Kapazitaet von 32 K Byte, falls im Slot 0 eine "Language Card" installiert ist sogar 48 K Byte. Sie belegt den Apple Speicher von \$4000 bis \$BFFF und, falls vorhanden, die gesamte "Language Card". Der Speicher wird von CP/M von "oben nach unten" belegt, d. h. zuerst die "Language Card" und dann den Apple Speicher von \$BFFF abwaerts bis \$4000. Die Initialisierung der Pseudodiskette erfolgt erst bei der 1. Benutzung, sie ist in 32 bzw. 48 Spuren zu je 1 K Byte eingeteilt.

**Hinweis:** Wird die Pseudodiskette (CP/M Laufwerk P:) zusammen mit dem Autoexecute Programm (AEXEC.COM) benutzt, so tritt beim Versuch die letzten 1 K Byte (Bereich \$4000 bis \$43FF) zu belegen ein BIOS Schreibschutz Fehler auf!

### 3.4 BIOS Fehlermeldungen

Neben den Fehlermeldungen des CP/M Systems (BDOS Error) gibt es im AP 22 spezielle BIOS-Fehlermeldungen, die dem Benutzer detaillierter ueber den aufgetretenen Fehler informieren und ihm zusaetzliche Reaktionsmoeglichkeiten bieten. BIOS Fehler treten grundsaeztlich nur bei nicht erfolgreichen Disketten Ein- oder Ausgaben auf, wie z.B. beim Schreibversuch auf eine schreibgeschuetzte Diskette, bei defekter Formatierung oder bei Datenfehlern. Die jeweiligen Meldungen und die Reaktionsmoeglichkeiten haengen vom Treiberprogramm des Disketten Laufwerks ab.

Eine BIOS - Fehlermeldung ist folgendermassen aufgebaut:

BIOS: <Fehlerart> Fehler auf <Laufwerk>:

<Fehlerart> ist die kurze Beschreibung des Fehlers,  
<Laufwerk> ist das CP/M Laufwerk auf dem der Fehler aufgetreten ist.

Nach Ausgabe der Fehlermeldung erwartet das BIOS die Eingabe eines Zeichens von der Console (Logisches CP/M Geraet CON:)

An dieser Stelle sind, abhaengig von der Fehlermeldung, einige oder alle der folgenden Angaben moeglich:

Eingabe	Bedeutung	Beschreibung
N	Noch einmal:	Es wird noch einmal versucht, die Disketten Ein- oder Ausgabe durchzufuehren, ist das nicht moeglich, so wird die Fehlermeldung erneut ausgegeben.
W	Weiterreichen:	Der aufgetretene Fehler wird an das aufrufende Programm weitergeleitet, die weitere Fehlerbehandlung erfolgt dort. War das aufrufende Programm z. B. das BDOS, so erfolgt die Meldung BDOS Err on <Laufwerk>: Bad Sector
I	Ignorieren:	Der aufgetretene Fehler wird ignoriert, d.h. obwohl ein Fehler aufgetreten ist, wird an das aufrufende Programm der Status "Kein Fehler" weitergegeben. Diese Moeglichkeit darf nur mit ausserster Vorsicht benutzt werden, u. U. kann dies, z.B. wenn ein Fehler, der beim Laden eines Programmes aufgetreten ist ignoriert wird, zu einem "Systemzusammenbruch" fuehren!
B	Boot:	Das laufende Programm wird abgebrochen, ein Warmstart ausgefuehrt und der CCP gestartet. Sie kehren mit dieser Eingabe in die Kommandoebene des CCP zurueck.

K Kaltstart: Wie B, aber zusaetzlich wird immer Laufwerk A: als Arbeitslaufwerk selektiert, der CCP meldet sich in jedem Fall mit A>

Die hier angegebenen Eingabemoeglichkeiten muessen nicht bei allen Fehlermeldungen zulassig sein. Bei einer falschen Eingabe werden die erlaubten Eingaben und anschliessend erneut die Fehlermeldung auf der Console ausgegeben.

Im einzelnen existieren die folgenden Fehlermeldungen:

fuer den Apple ][ Disk Controller:

Fehlerart	Bedeutung
Format	Die Formatierung der Diskette konnte nicht gelesen werden. Dieser Fehler tritt beim Lesen oder Schreiben auf nicht formatierte Disketten auf, er kann durchFormatierender Diskettebehoebenwerden. Andere Ursachen koennen sein:  stark abgenutzte Disketten, in diesem Fall sollte sie nicht wieder verwendet werden,  Schreibfehler des Disk II Controllers; solche Fehler koennen durch neu formatieren behoben werden
Positionier	Die ausgewaehlte Disketten-Spur konnte nicht gefunden werden, die Ursache hierfuer ist in der Regel eine fehlerhafte Formatierung oder aber ein Hardware Fehler im Positionier-Mechanismus des Laufwerks.
Hardware	Das angegebene Laufwerk ist entweder nicht vorhanden oder aber Disk Controller und/oder Laufwerk ist/sind defekt. Der Fehler kann auch auftreten, wenn im Laufwerk keine Diskette eingelegt ist, in diesem Fall wird aber normalerweise der Format Fehler ausgegeben.
Daten	Die Pruefsumme der gelesenen Daten eines Sektors stimmt nicht mit dem angegebenen Wert ueberein, d.h. die gelesenen Daten enthalten Fehler und koennen nicht mehr verwendet werden. Die Formatierung der Diskette ist in diesem Fall in Ordnung.
Schreibschutz	Die Diskette ist schreibgeschuetzt.

fuer die interne Pseudodisk und die AP 17 Pseudodisk:

**Positionier** Die angewaehlte Spur ist nicht vorhanden

**Daten** Nach Schreiben oder Lesen von Daten in bzw. aus der Pseudodisk werden diese verglichen. Stimmen die Daten nicht ueberein, so erfolgt diese Fehlermeldung.

nur fuer die interne Pseudodisk:

**Schreibschutz** Der von der Pseudodisk belegte Speicher wird auch vom AUTO EXECUTE Programm (AEXEC.COM) benutzt. Beim Versuch in Pseudodisk-Bereiche zu schreiben, die von diesem Programm belegt worden sind, wird dieser Fehler ausgegeben. In diesem Fall ist die interne Pseudodiskette voll.

### 3.5 Warmboot und Reset

Bei einem Warmstart (Warmboot) des CP/M-80 Vers. 2.2 Betriebssystems ist es erforderlich, den CCP und BDOS Teil neu in den Speicher zu laden. Ein solcher Warmstart erfolgt bei Betaetigen von CTRL-C in der CCP-Ebene oder, in der Regel, beim Verlassen eines Programmes. Da der CCP und BDOS Teil sich auf den Boot-Spuren der Diskette im Laufwerk A: befinden muss, um ihn von dort nachladen zu koennen, war es z.B. bei Systemen mit 1 Laufwerk erforderlich, dass sich das System auf allen Disketten befindet. Im AP 22 BIOS wird der CCP und BDOS Teil im HGR1-Bereich (Adr. \$2000-\$BFFF) des Apple Speichers zwischengespeichert und von dort nachgeladen, solange dieser Bereich nicht fuer andere Zwecke (z.B. High Resolution Grafik) benutzt wird. Aus diesem Grund ist es im AP 22 CP/M System nur in Ausnahmefaellen notwendig, dass sich das System auf den Boot Spuren befinden muss.

Beim Betaetigen der RESET-Taste (bzw. <CTRL> und <RESET> gleichzeitig), wird vom AP 22 BIOS der gesamte Z-80 Teil, des BIOS, Ausnahme der Seite FF, von der im Boot Laufwerk (normalerweise A:) eingelegten Diskette nachgeladen.

In diesem Fall muessen CCP, BDOS und BIOS auf den Boot Spuren dieser Diskette vorhanden sein!

Das CP/M System kann also durch Betaetigen von RESET neu gestartet werden, ohne das z.B. Daten der Pseudodiskette verloren gehen. Wird jedoch die Seite FF des AP 22 Speichers (Adr. 0FF00H bis 0FFFFH) oder der 6502 Teil des BIOS veraendert, so ist dies nicht mehr moeglich. In diesem Fall kann das CP/M System nur durch Aus- und Wiedereinschalten des Rechners neu gestartet werden; Daten die sich in einer Pseudodiskette befinden, gehen dabei natuerlich verloren!

### 3.6 Kommunikation zwischen AP 22 und Apple IC

Bedingt durch die Hardware der AP 22 kann die Z80 B CPU nicht direkt auf den Apple IC Speicher zugreifen. Die einzige Moeglichkeit den 6502 direkt zu beeinflussen ist die Erzeugung des IRQ-Signals, von der das AP 22 BIOS Gebrauch macht. Um dem Benutzer trotzdem auf einfache Weise Zugriff auf den Apple Speicher und auf 6502 Unterprogramme zu verschaffen, sind die Z80 Restart Befehle RST 1 bis RST 4 mit speziellen Kommunikationsfunktionen belegt worden. Diese Restart Befehle werden vom AP22 BIOS nicht benutzt und koennen daher bei Bedarf ueberschrieben werden, so dass dadurch keine Schwierigkeiten mit CP/M Programmen, die diese Befehle benutzen, entstehen koennen. Das AP22 BIOS traegt die entsprechenden Befehle bei jedem Warm- und Kaltstart, vor Rueckkehr in den CCP, wieder neu ein. Diese Befehle duerfen nur in Realzeit ausgefuehrt werden, die Benutzung von Trace oder Untrace Befehlen in einem Debugger, wie z. B. DDT, kann zum Systemzusammenbruch fuehren!

Die folgenden Funktionen sind eingebaut:

1. Aufruf eines 6502 Unterprogramms
2. Lesen und Schreiben einer Apple IC Speicherzelle
- und 3. Kopieren von Speicherbereichen vom AP 22 Speicher in den Apple Speicher und umgekehrt.

Die Restart Befehle 1 bis 3 sind mit Einzelfunktionen belegt, waehrend Restart 4 mehrere Funktionen besitzt. Auf den folgenden Seiten werden die Funktionen einzeln erlautert. Es werden die mnemotechnischen Abkuerzungen fuer die INTEL 8080 und die ZIL06 Z80 Schreibweise angegeben.

3.6.1 RESTART 18080: RST 1I80: RST 8Aufruf eines 6502 UnterprogrammsFunktion:

Mit diesem Befehl wird ein 6502 Unterprogramm aufgerufen. Die Adresse wird unmittelbar hinter dem Befehl in der Reihenfolge niederwertiges, hoehwertiges Byte angegeben. Das BIOS ruft das 6502 Unterprogramm mit der angegebenen Adresse auf, ausserdem werden die Inhalte der AP22 Speicherzellen OFFA4H bis OFFA7H in den Registern uebergeben (s. u.). Das Unterprogramm muss mit dem RTS Befehl beendet werden.

Aufruf:

```
8080: RST 1
      DW ADR$6502
      ...
```

```
I80: RST 8
     DEFW ADR_6502
     ...
```

Beschreibung:

Es wird das 6502 Unterprogramm ADR\$6502 bzw. ADR\_6502 aufgerufen. Die Registerinhalte des I80 werden durch den Aufruf nicht veraendert.

Stackbelastung:

Es werden 10 Byte des Stapelspeichers (ohne Aufruf) benoetigt.

Registeruebergabe:

Die Register des 6502 enthalten beim Unterprogrammaufruf die Inhalte der nachfolgend angegebenen AP22 Speicherzellen; nach Ausfuehrung des Unterprogramms werden die veraenderten 6502 Registerinhalte in die Speicherzellen OFF04H bis OFF07H geschrieben.

AP 22 Adresse (Aufruf)	6502 Register	AP 22 Adresse (Rueckgabe)
OFFA4H	Status	OFF04H
OFFA5H	Akku	OFF05H
OFFA6H	X	OFF06H
OFFA7H	Y	OFF07H

**HINWEIS:**

Das Interrupt Bedingungsbit des 6502 Status Registers kann nicht beeinflusst werden.

Die zurueckgegebenen Werte werden beim naechsten BIOS, B00S oder Restart Aufruf mit anderen Werten ueberschrieben! Sie muessen daher vorher ausgewertet oder aber gerettet werden.

3.6.2 RESTART 28080: RST 2Z80: RST 10HLesen einer 6502 SpeicherzelleFunktion:

Mit diesem Befehl wird eine 6502 Speicherzelle gelesen. Die Adresse wird unmittelbar hinter dem Befehl in der Reihenfolge niederwertiges, hoehwertiges Byte angegeben. Das BIOS liest die Speicherzelle und uebergibt deren Wert im A-Register des Z80.

Aufruf:

```
8080: RST 2
      DW MEM$6502
      ...
```

```
Z80: RST 10H
     DEFW MEM_6502
     ...
```

Beschreibung:

Es wird die 6502 Speicherzelle MEM\$6502 bzw. MEM\_6502 gelesen. Der Inhalt steht nach Ausfuehrung des Befehls im A-Register des Z80. Die anderen Register des Z80 bleiben unveraendert.

Stackbelastung:

Es werden 14 Byte des Stapelspeichers (ohne Aufruf) benoetigt.

3.6.3 RESTART 38080: RST 3Z80: RST 18HSchreiben in eine 6502 SpeicherzelleFunktion:

Mit diesem Befehl wird der Inhalt des A-Registers der Z80 CPU in eine 6502 Speicherzelle geschrieben, deren Adresse unmittelbar hinter dem Befehl in der Reihenfolge niederwertiges, hoherwertiges Byte angegeben wird. Die Z80 Registerinhalte bleiben unverändert.

Aufruf:

```
8080: MVI A,55H
      RST 3
      DW MEM$6502
      ...
```

```
Z80: LD A,55H
      RST 18H
      DEFW MEM_6502
      ...
```

Beschreibung:

Der Wert 55H (Hex 55) wird in die 6502 Speicherzelle MEM\$6502 bzw. MEM\_6502 geschrieben. Die Register des Z80 werden durch den Restart Befehl nicht verändert.

Stackbelastung:

Es werden 14 Byte des Stapelspeichers (ohne Aufruf) benötigt.

3.6.4 RESTART 48080: RST 4Z80: RST 20H

Speichertransfer, Lesen und Schreiben einer Speicherzelle  
sowie indirekter 6502 Unterprogrammaufruf

Beim Restart 4 Befehl werden durch das darauffolgende Byte verschiedene Funktionen angegeben. Im Gegensatz zu den Restart Befehlen 1 bis 3 werden hier die Adressen in den Z80 Registern angegeben.

8000,00 00  
8000,01 01  
8000,02 02  
8000,03 03  
8000,04 04  
8000,05 05  
8000,06 06  
8000,07 07  
8000,08 08  
8000,09 09  
8000,0A 0A  
8000,0B 0B  
8000,0C 0C  
8000,0D 0D  
8000,0E 0E  
8000,0F 0F

8000,10 10  
8000,11 11  
8000,12 12  
8000,13 13  
8000,14 14  
8000,15 15  
8000,16 16  
8000,17 17  
8000,18 18  
8000,19 19  
8000,1A 1A  
8000,1B 1B  
8000,1C 1C  
8000,1D 1D  
8000,1E 1E  
8000,1F 1F

8000,20 20  
8000,21 21  
8000,22 22  
8000,23 23  
8000,24 24  
8000,25 25  
8000,26 26  
8000,27 27  
8000,28 28  
8000,29 29  
8000,2A 2A  
8000,2B 2B  
8000,2C 2C  
8000,2D 2D  
8000,2E 2E  
8000,2F 2F

8000,30 30  
8000,31 31  
8000,32 32  
8000,33 33  
8000,34 34  
8000,35 35  
8000,36 36  
8000,37 37  
8000,38 38  
8000,39 39  
8000,3A 3A  
8000,3B 3B  
8000,3C 3C  
8000,3D 3D  
8000,3E 3E  
8000,3F 3F

8000,40 40  
8000,41 41  
8000,42 42  
8000,43 43  
8000,44 44  
8000,45 45  
8000,46 46  
8000,47 47  
8000,48 48  
8000,49 49  
8000,4A 4A  
8000,4B 4B  
8000,4C 4C  
8000,4D 4D  
8000,4E 4E  
8000,4F 4F

3.6.4.0 RESTART 4/08080: RST 4  
DB 0Z80: RST 20H  
DEFB 0Speichertransfer Apple IC -> AP 22Funktion:

Beginnend mit der Adresse im HL-Register wird eine Apple Speicherzelle gelesen und in die entsprechende im Register DE angegebene AP 22 Adresse geschrieben. Anschliessend werden HL und DE-Register um 1 erhöht und vom BC-Register 1 subtrahiert. Ist BC danach nicht 0, so wird das ganze wiederholt. Die Abarbeitung dieses Befehls gleicht der des Z80 LDIR-Befehls.

Z80 Registerinhalte vor Ausfuehrung des Befehls:

HL - Anfangsadresse im Apple IC Speicher

DE - Zieladresse im AP 22 Speicher

BC - Laenge des Speichertransfers in Byte

Z80 Registerinhalte nach Ausfuehrung des Befehls:

HL - Adresse des letzten aus dem Apple IC Speicher uebertragenen Bytes + 1

DE - Adresse des letzten in den AP 22 Speicher uebertragenen Bytes + 1

BC - 0

Das AF-Register und die Zweitregister bleiben unveraendert.

Aufruf:8080: LXI H,2000H  
LXI D,4000H  
LXI B,2000H  
RST 4  
DB 0  
...Z80: LD HL,2000H  
LD DE,4000H  
LD BC,2000H  
RST 20H  
DEFB 0  
...Beschreibung:

Der HGR1-Bereich im Apple Speicher (\$2000-\$3FFF) wird in den AP22 Speicher ab Adresse 4000H bis 5FFFH uebertragen. Nach Ausfuehrung des Befehls enthaelt HL 4000H, DE 6000H und BC 0.

Stackbelastung:

Es werden 12 Byte Stapelspeicher (ohne Aufruf) benoetigt.

3.6.4.1 RESTART 4/18080: RST 4  
DB 1Z80: RST 20H  
DEFB 1Speichertransfer AP 22 -> Apple ICFunktion:

Beginnend mit der Adresse im HL-Register wird eine AP 22 Speicherzelle gelesen und in die entsprechende im Register DE angegebene Apple Speicherzelle geschrieben. Anschliessend werden HL und DE-Register um 1 erhöht und vom BC-Register 1 subtrahiert. Ist BC danach nicht 0, so wird das ganze wiederholt. Die Abarbeitung dieses Befehls gleicht der des Z80 LDIR-Befehls.

Z80 Registerinhalte vor Ausfuehrung des Befehls:

HL - Anfangsadresse im AP 22 Speicher

DE - Zieladresse im Apple IC Speicher

BC - Laenge des Speichertransfers in Byte

Z80 Registerinhalte nach Ausfuehrung des Befehls:

HL - Adresse des letzten aus dem AP 22 Speicher uebertragenen Bytes + 1

DE - Adresse des letzten in den Apple IC Speicher uebertragenen Bytes + 1

BC = 0

Das AF-Register und die Zweitregister bleiben unveraendert.

Aufruf:8080: LXI H,1000H  
LXI D,2000H  
LXI B,2000H  
RST 4  
DB 1  
...Z80: LD HL,1000H  
LD DE,2000H  
LD BC,2000H  
RST 20H  
DEFB 1  
...Beschreibung:

Der Speicherbereich 1000H bis 2FFFH des AP 22 Speichers wird in den HGRI-Bereich des Apple Speichers (\$2000-\$3FFF) kopiert. Nach Ausfuehrung des Befehls enthaelt HL 3000H, DE 4000H und BC 0.

Stackbelastung:

Es werden 12 Byte Stapelspeicher (ohne Aufruf) benoetigt.

3.6.4.2 RESTART 4/2

8080: RST 4  
DB 2

Z80: RST 20H  
DEFB 2

Lesen einer 6502 SpeicherzelleFunktion:

Mit diesem Befehl wird eine 6502 Speicherzelle gelesen. Die Adresse wird im HL-Register des Z80 angegeben. Das BIOS liest die Speicherzelle und uebergibt deren Inhalt im A-Register des Z80.

Aufruf:

```
8080: LXI H, MEM$6502      Z80: LD   HL, MEM_6502
      RST 4                RST 20H
      DB 2                  DEFB 2
      ...                  ...
```

Beschreibung:

Es wird die 6502 Speicherzelle MEM\$6502 bzw. MEM\_6502 gelesen. Der Inhalt steht nach Ausfuehrung des Befehls im A-Register des Z80. Die anderen Register des Z80 bleiben unveraendert.

Stackbelastung:

Es werden 14 Byte des Stapelspeichers (ohne Aufruf) benoetigt.

3.6.4.3 RESTART 4/3

8080: RST 4  
DB 3

Z80: RST 20H  
DEFB 3

Schreiben in eine 6502 SpeicherzelleFunktion:

Mit diesem Befehl wird der Inhalt des A-Registers der Z80 CPU in eine 6502 Speicherzelle geschrieben, deren Adresse im HL-Register angegeben wird. Die Z80 Registerinhalte bleiben unverändert.

Aufruf:

8080: LXI H, MEM\$6502  
MVI A, 46H  
RST 4  
DB 3  
...

Z80: LD HL, MEM\_6502  
LD A, 46H  
RST 20H  
DEFB 3  
...

Beschreibung:

Der Wert 46H (Hex 46) wird in die 6502 Speicherzelle MEM\$6502 bzw. MEM\_6502 geschrieben. Die Register des Z80 werden durch den Restart Befehl nicht veraendert.

Stackbelastung:

Es werden 14 Byte des Stapelspeichers (ohne Aufruf) benoetigt.

3.6.4.4 RESTART 4/4

8080: RST 4  
DB 4

Z80: RST 20H  
DEFB 4

Aufruf eines 6502 UnterprogrammsFunktion:

Mit diesem Befehl wird ein 6502 Unterprogramm aufgerufen. Die Adresse wird im HL-Register des Z80 angegeben. Das BIOS ruft das 6502 Unterprogramm mit der angegebenen Adresse auf, ausserdem werden die Inhalte der AP22 Speicherzellen OFFA4H bis OFFA7H in den Registern uebergeben (s. u.). Das Unterprogramm muss mit dem RTS Befehl beendet werden.

Aufruf:

8080: LXI H,ADR\$6502  
RST 4  
DB 4  
...

Z80: LD HL,ADR\_6502  
RST 20H  
DEFB 4  
...

Beschreibung:

Es wird das 6502 Unterprogramm ADR\$6502 bzw. ADR\_6502 aufgerufen. Die Registerinhalte des Z80 werden durch den Aufruf nicht veraendert.

Stackbelastung:

Es werden 10 Byte des Stapelspeichers (ohne Aufruf) benoetigt.

Registeruebergabe:

Die Register des 6502 enthalten beim Unterprogrammaufruf die Inhalte der nachfolgend angegebenen AP22 Speicherzellen; nach Ausfuehrung des Unterprogramms werden die veraenderten 6502 Registerinhalte in die Speicherzellen OFF04H bis OFF07H geschrieben.

AP 22 Adresse (Aufruf)	6502 Register	AP 22 Adresse (Rueckgabe)
OFFA4H	Status	OFF04H
OFFA5H	Akku	OFF05H
OFFA6H	X	OFF06H
OFFA7H	Y	OFF07H

**HINWEIS:**

Das Interrupt Bedingungsbit des 6502 Status Registers kann nicht beeinflusst werden.

Die zurueckgegebenen Werte werden beim naechsten BIOS, BDOS oder Restart Aufruf mit anderen Werten ueberschrieben! Sie muessen daher vorher ausgewertet oder aber gerettet werden.

#### 4. BIOS Details

In diesem Abschnitt finden Sie in erster Linie Informationen ueber das BIOS (Basic Input/Output System) des AP 22 CP/M Systems. Diese benoetigen Sie zur Anpassung an andere Peripheriekarten oder zur Implementierung anderer Disk Controller. In diesem Abschnitt werden gute Kenntnisse der 8080/Z80 und der 6502 Assemblersprache vorausgesetzt.

##### 4.1 Arbeitsweise des BIOS

Die AP 22 stellt einen eigenstaendigen vom Apple II steuerbaren Z80 8 Rechner mit eigenen 64K Byte Speicher dar. Die Z80 CPU kann nur innerhalb des AP 22 Speichers arbeiten; sie kann nicht auf den Apple Speicher zugreifen. Daraus ergibt sich das alle Ein- und Ausgaben durch 6502 Treiber erfolgen muessen. Um einen solchen Treiber aufzurufen sendet der Z80 einen IRQ an den 6502 zusammen mit einer Treibernummer und 6 Byte Parameter. Anschliessend wartet der Z80 auf die Beendigung des Treiberaufrufs indem er ein Synchronisierungsbyte abfragt in welches der 6502 eine Endekennung eintragen muss. Der 6502 kopiert, nach Aktivierung durch den IRQ, zunaechst die Treibernummer und die Parameter in 7 Bytes der Zero Page (\$F9 bis \$FF). Ist Bit 7 der Treibernummer gesetzt (d. h. der Treiber soll parallel zum Z80 ablaufen) so wird schon jetzt die Endekennung im Synchronisierungsbyte eingetragen so dass der Z80 weiterlaufen kann. Jetzt wird der eigentliche Treiber aufgerufen. Da der Z80 nur eine 8 Bit Treibernummer uebergibt, befindet sich im Apple Speicher eine Adresstabelle ab Adresse \$000 in der fuer jede Treibernummer eine 16 Bit Adresse angegeben ist (\$000 (LSB), \$001 (MSB) = Adresse fuer Treiber 0; \$002, \$003 = Adr. Treiber 1; usw.). Bit 7 der Treibernummer wird dabei nicht beruecksichtigt. Durch dieses Verfahren braucht der Z80 Teil des BIOS nicht mit 6502 Adressen zu arbeiten. Die sich ergebende Treiberadresse wird mit einem JSR-Befehl aufgerufen. Fuer den Treiber selbst stehen die 6 Parameter Bytes zur Verfuegung. Diese duerfen beliebig veraendert werden. In welchen Parametern was angegeben ist finden Sie in den Treiber Spezifikationen. Ist der Treiber abgearbeitet so wird im Synchronisationsbyte die Endekennung eingetragen, zusaetzlich werden, allerdings nur wenn Bit 7 der Treibernummer nicht gesetzt war (sequentielle Abarbeitung), die (ggf. modifizierten) Parameter in den AP 22 Speicher zurueckkopiert. Bei sequentieller Abarbeitung kann der Z80 diese Werte anschliessend auswerten.

Nach diesem Schema werden alle 6502 Treiber aufgerufen. Welche Werte dem jeweiligen Treiber in den Parameter Bytes uebergben und welche, bei sequentieller Abarbeitung, zurueckgegeben werden finden Sie in den Abschnitten 4.4 und 4.5.

#### 4.2 Speicherbelegung

Das AP 22 BIOS belegt sowohl Teile des Apple IC wie auch des AP 22 Speichers. Zunaechst der 280 Bereich (alle Adressen Hexadezimal):

Adresse(n)	Bedeutung
0000 - 0002	Warmstart Einsprung
0003	IOBYTE (Zuordnung logische -> phys. Geraete)
0004	aktuelles Default Laufwerk
0005 - 0007	BDOS Einsprung
0008 - 000A	Restart 1 Einsprung
0010 - 0014	Restart 2 Einsprung
0018 - 001F	Restart 3 Einsprung
0020 - 0022	Restart 4 Einsprung
0038 - 003A	Reserviert fuer Debugger
0040 - 0048	Reserviert
0049 - 004C	Video Funktions Ausgabe
005C - 006B	Default FCB 1
006C - 007B	Default FCB 2
0080 - 00FF	CCP Kommando Zeile / Default DMA Buffer
0100 - 0FFF	TPA
E000 - E7FF	CCP
E800 - F5FF	BDOS
F600 - ca. FC37	BIOS Code und Tabellen
ca. FC38 - (MEMEND)	BIOS nicht initialisierte Daten (s. Seite FF) Der Wert (MEMEND) befindet sich in der Seite FF des AP 22 Speichers und giebt die oberste belegte Adresse des Bereichs an, der von Anzahl und Kapazitaet der verwendeten Disk Laufwerke abhaengig ist. Deren Wert darf nicht bis in den internen BIOS Stack reichen. Der Bereich zwischen (MEMEND) und dem Stack ist frei.
ca. FDAB - FDCF	interner BIOS Stack
FDE0 - FDFF	nicht initialisierte Disk Variable
FE00 - FEFF	256 Byte Disk Buffer

Der Bereich 0FF00H bis 0FFFFH des AP22 Speicher hat eine besondere Bedeutung. Hier befinden sich Speicherzellen zur Kommunikation mit dem 6502 und Konfigurationstabellen. Dieser Bereich wird nur beim Kaltstart initialisiert, während dies beim BIOS Code auch beim Betaetigen von <RESET> geschieht. Hier die Belegung:

Adresse(n)	Bedeutung
FF00	Synchronisations Byte fuer Treiber Aufrufe
FF01	Treibernummer
FF02	1. Treiber Parameter (LSB)
FF03	1. Treiber Parameter (MSB)
FF04	2. Treiber Parameter (LSB)
FF05	2. Treiber Parameter (MSB)
FF06	3. Treiber Parameter (LSB)
FF07	3. Treiber Parameter (MSB)
FF08	Apple Tastatur Status (FF = Zeichen bereit)
FF09 - FF0F	reserviert
FF10	Anzahl implementierter Laufwerke (ohne P:)
FF11	Peripheriekartentyp Slot 1
FF12	Peripheriekartentyp Slot 2
FF13	Peripheriekartentyp Slot 3
FF14	Peripheriekartentyp Slot 4
FF15	Peripheriekartentyp Slot 5
FF16	Peripheriekartentyp Slot 6
FF17	Peripheriekartentyp Slot 7
FF18 - FF1C	Disk Konf. Tabelle Laufwerk A:
FF1D - FF21	Disk Konf. Tabelle Laufwerk B:
FF22 - FF26	Disk Konf. Tabelle Laufwerk C:
FF27 - FF2B	Disk Konf. Tabelle Laufwerk D:
FF2C - FF30	Disk Konf. Tabelle Laufwerk P:

Adresse(n)	Bedeutung
FF31 - FF40	Disk Parameter Header (DPH) Laufwerk A:
FF41 - FF50	DPH Laufwerk B:
FF51 - FF60	DPH Laufwerk C:
FF61 - FF70	DPH Laufwerk D:
FF71 - FF80	DPH Laufwerk P:
FF81 - FF90	reserviert fuer neuen DPB
FF91 - FF94	Disk Ergaenzungs Tabelle fuer Apple IC Disk
FF95 - FF98	Disk Erg. Tabelle fuer AP 17 Pseudodisk
FF99 - FF9C	Disk Ergaenzungs Tabelle alt. Disk Controller
FF9D - FFA0	Disk Ergaenzungs Tabelle fuer Harddisk
FFA1	belegt
FFA2	Flag: 0 = keine BIOS Fehlermeldungen
FFA3	belegt
FFA4	6502 Unterprg. Aufruf: Status
FFA5	6502 Unterprog. Aufruf: Akku
FFA6	6502 Unterprog. Aufruf: X-Register
FFA7	6502 Unterprog. Aufruf: Y-Register
FFA8 - FFAF	frei
FFB0 - FFBF	Kopie von Adr. 40H bis 4FH
FFC0 - FFCB	frei
FFC9	belegt
FFCA - FFCB	MEMEND: Speicherende (LSB, MSB)
FFCC - FFCF	belegt
FFCE - FFCF	Anfangsadresse der Geraete Treibernummern Tab.
FFD0 - FFEF	Kopie von Adr. 8 bis 27H
FFF0 - FFFF	frei

Tabelle 6: AP 22 Speicherbelegung

## Die Belegung des Apple Speichers:

Adresse(n)	Bedeutung
0	Hilfsvariable
6	AP 22 Slot * 16
7 - 9	belegt
19 - 1A	belegt
1B	Hilfsvariable
26 - 27	belegt
2A - 2B	belegt
2D - 2E	belegt
3D - 3F	belegt
45 - 47	belegt
CE - CF	belegt
F9	Treibernummer
FA	1. Treiber Parameter (LSB)
FB	1. Treiber Parameter (MSB)
FC	2. Treiber Parameter (LSB)
FD	2. Treiber Parameter (MSB)
FE	3. Treiber Parameter (LSB)
FF	3. Treiber Parameter (MSB)

Tabelle 7: 6502 Zero Page Belegung

Adresse(n)	Bedeutung
100 - 1FF	6502 Stack
200 - 3EF	frei
3F0 - 3FF	Interrupt und Reset Vektoren
400 - 7FF	Apple 24x40 Zeichen Speicher
800 - BFF	nur in der Kaltstart Phase belegt
C00 - 1AAF	6502 Teil des AP 22 BIOS
1AB0 - 1AFF	Patch Bereich
1B00 - 1C55	Disk Buffer fuer Apple ][ Disk
1C56 - 1EFF	frei
1F00 - 1FFF	nur in der Kaltstart Phase belegt
2000 - 3FFF	High Resolution Grafik 1 Bereich enthaelt CCP und B00S Teil des CP/M Betriebssystems zum Nachladen beim Warmstart
4000 - BFFF	Interne Pseudodisk (Laufwerk P:) und Speicher fuer Autoexecute Kommandos. Die Pseudodisk belegt den Speicher von BFFF abwaerts, waehrend die Autoexecute Kommandos von 4000 aufwaerts abgespeichert werden.

**Tabelle 8:** Apple ][ Speicherbelegung

Die hier als "frei" gekennzeichneten Bereiche koennen von zusaetzlichen Treibern oder Benutzerprogrammen belegt werden, sie werden vom AP 22 BIOS nicht benutzt.

"nur in der Kaltstart Phase belegt" bedeutet das diese Bereiche nur unmittelbar nach dem Einschalten vom AP 22 BIOS benutzt werden; sie koennen ebenfalls von Benutzerprogrammen belegt werden.

Mit "belegt" gekennzeichnete Speicherzellen duerfen nicht benutzt werden.

"Hilfsvariable" duerfen von zusaetzlichen Treibern temporaer benutzt werden.

### 4.3 Konfigurations Tabellen

Das AP 22 BIOS arbeitet mit einer automatischen Peripheriekartenerkennung in der Kaltstart Phase. Fuer Karten mit eingebautem ROM (Peripheral Card ROM Space) existiert eine spezielle Karten Erkennungs Tabelle in der bis zu 3 beliebige Bytes zur Erkennung einer Karte definiert werden koennen. Zusaetzlich sind dort auch die zugehoerige Kennung der Karte fuer die Slot-Tabelle (Adr. 0FF09 (Slot 1) bis 0FF0F (Slot 7)) im AP 22 Speicher) und die Basis Treibernummer fuer den Peripheriekarten Treiber angegeben (dazu mehr in den Abschnitten 4.4 und 4.5).

Ein Eintrag in dieser Tabelle sieht wie folgt aus:

Byte 0: Basis Treibernummer  
 Byte 1: Kennung in der Slot-Tabelle  
 Byte 2: 3. Kennungs Byte  
 Byte 3: Adresse des 3. Kennungs Bytes im ROM  
 Byte 4: 2. Kennungs Byte  
 Byte 5: Adresse des 2. Kennungs Bytes im ROM  
 Byte 6: 1. Kennungs Byte  
 Byte 7: Adresse des 1. Kennungs Bytes im ROM

Einige Slot Tabellen Kennungen sind bereits festgelegt:

Kennung	Peripheriekarte
0	keine Peripheriekarte oder unbek. Karte ohne ROM
1	unbekannte Peripheriekarte mit ROM
2	AP 22
3	Serial Communication Interface
4	80 Zeichen Karte
5	Parallel Printer Interface
80H	Apple II Disk Controller
81H	AP 17 o. ae. Pseudodisk
82H	andere Floppy Disk Controller
83H	Harddisk
90H	nicht benutzte Speicherkarte mit 64K RAM
91H	nicht benutzte Speicherkarte mit 128K RAM
92H	nicht benutzte Speicherkarte mit 196K RAM
93H	nicht benutzte Speicherkarte mit 256K RAM

Allgemein haben Disk Controller Kennungen Werte ab 80H und andere Peripheriekarten Werte von 3 bis 7FH.

Als Basis Treibernummern sind fuer Peripheriekarten die Werte \$20 bis \$3C in 4er Schritten, fuer Disk Controller \$40 bis \$52 in 6er Schritten, vorgesehen, und zwar:

\$40 fuer Apple IC Disk,  
 \$46 fuer AP 17 o. ae. und interne Pseudodisk,  
 \$4C fuer andere Floppy Disk Controller und  
 \$52 fuer Harddisk Controller.

Die Basis Treibernummern \$20 bis \$2F und \$40 bis \$4B sind bereits belegt.

Die Karten Erkennungs Tabelle hat insgesamt 6 Eintraege von denen derzeit 4 belegt sind. Diese Eintraege werden vom BIOS von hinten nach vorne ausgewertet. Wird fuer ein Erkennungsbyte als Adress- und Datenwert 0 angegeben, so wird es nicht beruecksichtigt; es koennen also auch weniger als 3 Kennungsbyte definiert werden. Lage und Inhalt der Standard Tabelle finden Sie im Anhang E.

Fuer Disk Controller gibt es zusätzlich eine "Disk Ergaenzungs Tabelle" als Erweiterung zur Erkennungstabelle (s. Abschnitt 4.5).

#### 4.4 Peripheriekarten Schnittstelle

Fuer jede Text Peripheriekarte muessen bis zu 4 6502 Treiberfunktionen zur Verfuegung stehen. Diese sind:

1. Initialisierung des Treibers bzw. der Peripheriekarte
2. Ein- oder Ausgabestatus (je nach Verwendung)
3. Einlesen eines Zeichens
4. Ausgabe eines Zeichens

Die Einsprungadressen fuer diese 4 Treiberfunktionen muessen in der Treiber Adress Tabelle, abhaengig von der Basis Treibernummer der Karte, eingetragen werden, und zwar an folgenden Stellen:

- |                     |  |
|---------------------|--|
| 1. Initialisierung: | $\$000 + \text{Basis Treibernummer} * 2$       |
| 2. Status:          | $\$000 + (\text{Basis Treibernummer} + 1) * 2$ |
| 3. Eingabe:         | $\$000 + (\text{Basis Treibernummer} + 2) * 2$ |
| 4. Ausgabe:         | $\$000 + (\text{Basis Treibernummer} + 3) * 2$ |

z. B. fuer Basis Treibernummer \$20:

Initialisierungsadresse:	LSB in \$040, MSB in \$041;
Status:	LSB in \$042, MSB in \$043;
Eingabe:	LSB in \$044, MSB in \$045;
Ausgabe:	LSB in \$046, MSB in \$047;

Die einzelnen Funktionen werden wie folgt aufgerufen:

##### 1. Initialisierung

Eingangsparameter: Slot-Nr. in \$FB  
Ausgangsparameter: keine

Die Initialisierung wird beim Kaltstart und bei jedem Warmstart aufgerufen.

##### 2. Status

Eingangsparameter: Slot-Nr. in \$FB  
Ausgangsparameter: Status in \$FD und zwar \$FF wenn das Gerat bereit ist ein Zeichen zu empfangen (Eingabe Status) oder auszugeben (Ausgabe Status), andernfalls muss \$00 zurueckgegeben werden.

Verfuegt die Karte nicht ueber eine Statusabfrage, so muss immer \$00 (nicht bereit) uebergeben werden. Normalerweise wird der Eingabestatus verwendet, lediglich bei Drucker Interfaces wird der Ausgabestatus uebergeben.

### 3. Eingabe

Eingangsparameter: Slot-Nr. in \$FB  
Ausgangsparameter: ASCII-Zeichen in \$FD

Verfügt die Peripheriekarte nicht ueber eine Eingabemöglichkeit, so muss \$1A (CTRL-Z) zurueckgegeben werden.

### 4. Ausgabe

Eingangsparameter: Slot-Nr. in \$FB  
Zeichen in \$FE

Das angegebene Zeichen muss ausgegeben werden. Bit 7 wird nicht gesetzt.

Die eingebauten Peripheriekarten Treiber finden Sie im Anhang A.

#### 4.5 Disketten Treiber Schnittstelle

Zur Anpassung eines neuen Disk Controllers an das AP 22 BIOS muessen Aufbau und Funktionsweise der Disk Parameter Header (DPH) und Disk Parameter Bloecke (DPB) unbedingt bekannt sein. Informationen darueber finden Sie im "CP/M 2.2 ALTERATION GUIDE", einem Teil der CP/M Dokumentation (in der SoftCard Dokumentation nicht enthalten) oder in deutscher Sprache im Buch "Vom Umgang mit CP/M" von Bernd Pol erschienen im IWT-Verlag Vaterstetten (ISBN 3-88322-004-3).

Im AP 22 BIOS sind Slot-Nr. und Art eines Disk Controllers variabel. Um dies zu erreichen verfuegt das BIOS ueber mehrere Tabellen die die Eigenschaften des Disk Controllers und die Zuordnung zum CP/M Laufwerk festlegen. Grundsaeztlich wird zwischen dem eigentlichen Disk Controller im Slot mit seinem Laufwerk und dem CP/M Laufwerk unterschieden. Wie das eine dem anderen zugeordnet ist kann vom Anwender festgelegt werden.

Zum Disk Controller gehoeren 2 Tabelleneintraege:

1. in der Slot Karten Erkennungs Tabelle, dieser Eintrag dient zur Erkennung des Disk Controllers und
2. in der Disk Ergaenzungs Tabelle.

Punkt 1. wird im Abschnitt 4.3 behandelt; die Disk Ergaenzungstabelle besteht aus 4 Eintraegen mit je 4 Byte Laenge. Welcher Eintrag verwendet wird ist von der Kennung in der Erkennungstabelle abhaengig. Die Tabelle befindet sich, waehrend der Kaltstart Phase, ab Adresse \$B91 bis \$BA0 im Apple IC Speicher. Ein Eintrag ist wie folgt aufgebaut:

Byte 0, Byte 1: Adresse des Disk Parameter Blocks (DPB) im AP 22 Speicher. Dieser Block legt die physikalischen Eigenschaften eines Laufwerks fest, wie z. B. Sektoren pro Spur, Datenkapazitaet, Anzahl der Systemspuren, usw. fest. Eine genaue Beschreibung finden Sie in der am Anfang des Abschnitts angegebenen Literatur.

Byte 2: Offset auf die Skew Tabelle oder 0 wenn sie nicht vorhanden ist. Die Anfangsadresse errechnet sich aus DPB-Adr. + Offset. Das BIOS spricht die Sektoren einer Spur normalerweise sequentiell an, d. h. zunaechst Sektor 0, dann 1, 2, 3 usw.. Wird eine Diskette in dieser Weise angesprochen so ergeben sich erhebliche Wartezeiten, da nach dem Lesen eines Sektors nahezu 1 Disketten Umdrehung gewartet werden muss bis der naechste Sektor gelesen werden kann. Der Grund dafuer ist der, dass nach dem Lesen eines Sektors etwas Bearbeitungszeit vergeht bevor der naechste

angefordert wird. In dieser Zeit hat dieser aber mittlerweile den Lesekopf des Laufwerks passiert. Um diesen Zeitverlust zu minimieren kann mit der Skew Tabelle die Reihenfolge der anzusprechenden Sektoren veraendert werden. Dort wird fuer jeden vom BIOS angeforderten (logischen) Sektor der tatsaechlich anzusprechende (physikalische) Sektor angegeben. In der Skew Tabelle des AP 22 BIOS werden die Nummern der 256(!) Byte Sektoren angegeben. Beispiele finden Sie in der angegebenen Literatur.

Byte 3:

Wiederholungsanzahl. Dieser Wert gibt an wie oft eine Schreib- oder Leseoperation wiederholt wird bevor ein (BIOS-) Fehler gemeldet wird.

Die gesamte Tabelle besteht aus 4 Eintraegen, von denen die ersten beiden bereits belegt sind:

Adresse des Eintrags	Kennung in Erkennungs Tabelle	Disk Controller Typ
\$B91 - \$B94	80H	Apple II Disk
\$B95 - \$B98	81H	AP 17 o. ae. Pseudodisk
\$B99 - \$B9C	82H	andere Floppy Controller
\$B9D - \$BAD	83H	Harddisk

Diese Tabelle wird vom AP 22 BIOS nur waehrend der Kaltstart Phase benutzt.

Zu einem Disk Controller gehoeren, unabh'aengig davon wieviel Laufwerke an ihn angeschlossen sind, ein Eintrag in der Karten Erkennungs Tabelle, ein Eintrag in der Disk Erg'aenzungs Tabelle, ein Disk Parameter Block und, optional, eine Skew Tabelle. Die Werte der Disk Erg'aenzungs Tabelle werden waehrend der Kaltstart Phase in den zum CP/M Laufwerk gehoerenden Tabellen eingetragen.

Zum CP/M Laufwerk gehoeren die Disk Konfigurations Tabelle und der Disk Parameter Header. Die Disk Konfigurations Tabelle befindet sich in der Kaltstart Phase ab Adresse \$B18 bis \$B30 im Apple, und spaeter ab Adresse 0FF18H bis 0FF30H im AP 22 Speicher. In der Tabelle ist fuer jedes CP/M Laufwerk ein Eintrag mit je 5 Byte Laenge vorhanden. Von diesen Bytes ist nur 1 zur Konfiguration bestimmt: Byte 1 legt den zugeordneten Slot und die Laufwerksnummer fest und wird in der Form Slot \* 16 + Laufwerk (Laufwerk ab 0 nummeriert) angegeben. Der gesamte Eintrag ist wie folgt aufgebaut:

- Byte 0: Basis Treiber Nummer fuer den zugehoerigen Disk Controller. Dieses Byte wird in der Kaltstart Phase, anhand des in der Karten Erkennungs Tabelle angegebenen Wertes, eingesetzt.
- Byte 1: Slot \* 16 + Laufwerk. Mit diesem Byte wird dem CP/M Laufwerk das Laufwerk eines Disk Controllers im angegebenen Slot zugeordnet. Ist dort kein Disk Controller installiert, so wird als Basis Treiber Nummer (Byte 0) 0FFF eingetragen. Dieses Byte muss vom Benutzer eingetragen werden.
- Byte 2: Wiederholungsanzahl. Dieser Wert wird vom BIOS anhand des in der Disk Ergaenzungs Tabelle fuer den Disk Controller angegebenen Wertes eingesetzt.
- Byte 3, 4: Hier merkt sich das BIOS die zuletzt angesprochene Spur des Laufwerks. Der Anfangswert muss 0FFFFH sein. Dieser Wert wird zur Rekalibrierung der Apple IC Disk und zur Initialisierung der AP 17 Pseudodisk verwendet.

Die gesamte Tabelle besteht aus 5 Eintraegen (incl. Pseudodisk P:) und ist wie folgt aufgebaut:

Adressen	CP/M Laufwerk
\$B18 - \$B1C	A:
\$B1D - \$B21	B:
\$B22 - \$B26	C:
\$B27 - \$B2B	D:
\$B2C - \$B30	P:

Fuer Laufwerk P: werden vom BIOS keine Werte eingesetzt, es ist immer definiert.

Fuer jedes CP/M Laufwerk steht je 1 Disk Parameter Header (DPH) zur Verfuegung, den genauen Aufbau finden Sie in der am Anfang angegebenen Literatur. Vom BIOS werden in der Kaltstart Phase zunaechst die Adressen XLT (Anfang der Skew Tabelle) und DPB (Disk Parameter Block) und anschliessend, im Z80 Teil der Kaltstart Phase, auch die Adressen fuer ALV und CSV, abhaengig vom jeweiligen DPB, eingetragen. Die dafuer notwendigen Speicherbereiche werden dem freien BIOS-Speicher entnommen, die Endadresse (zu finden in den Speicherzellen 0FFCAH und 0FFCBH) wird entsprechend hochgesetzt. Diese Adresse darf nach der Kaltstart Phase nicht bis in den BIOS Stack reichen! (s. Speicherbelegung)

Wie verwendet das BIOS diese Tabellen ?

Diese Frage soll durch den Ablauf der Diskettenerkennung und Zuordnung waehrend der Kaltstart Phase des BIOS erlaeuert werden.

Nachdem ein Disk Controller anhand seines ROM's erkannt worden ist (s. Abschnitt 4.3) wird seine Kennung zunaechst in die Slot Tabelle eingetragen. Nachdem alle Slots ueberprueft worden sind, benutzt das BIOS die Disk Konfigurations Tabelle um zu ermitteln welche CP/M Laufwerke fuer welche Slots definiert sind. Durch diese Tabelle wird die Verbindung zwischen dem Disk Controller mit seinem Laufwerk und der CP/M Laufwerksbezeichnung hergestellt. Ist dies geschehen so wird die Disk Ergaenzungs Tabelle verwendet um die zum Disk Controller gehoerenden Werte, wie DPB-, XLT - Adresse und Wiederholungsanzahl in die Tabellen des CP/M Laufwerks einzutragen. Als letztes werden, bereits im Z80 Teil der Kaltstart Phase, die ALV und CSV Bereiche fuer die Laufwerke A: bis D: belegt.

Nun zu den eigentlichen Treibern:

Insgesamt hat ein Disk Treiber 6 verschiedene Einsprungpunkte:

1. Deselektierung des Laufwerks
2. Selektierung des Laufwerks
3. Positionieren
4. Lesen eines (256 Byte) Sektors
5. Schreiben eines (256 Byte) Sektors und
6. Formatieren einer Spur (nur fuer Disk Manager)

Die Einsprungsadressen fuer diese Funktionen muessen in der Treiber Adress Tabelle beginnend mit der Basis Treibernummer eingetragen sein, d. h. bei Basis Treibernr. 40H in \$D0, \$D1 die Adresse fuer Deselektierung, ..., in \$D8A, \$D8B die Adresse fuer Formatieren einer Spur.

Den einzelnen Funktionsaufrufen werden Parameter in der Zero Page des 6502 uebergeben, diese sind:

1. Deselektieren des Laufwerks

Eingangsparameter: \$FA - Slot \* 16 + Laufwerk

Ausgangsparameter: keine

Mit dieser Funktion wird das in \$FA angegebene Laufwerk des Disk Controllers im ebenfalls dort angegebenen Slot deselektiert, d. h. es wird erst wieder nach einer erneuten Selektierung angesprochen.

2. Selektieren des Laufwerks

Eingangsparameter: \$FA - Slot \* 16 + Laufwerk  
\$FC, \$FD - letzte            zuvor  
                                  angesprochene Spur (LSB,  
                                  MSB)

Ausgangsparameter: \$FB - Fehlerkennung (0=kein Fehler)  
Status - Carry Clear = Fehler

falls Fehlerkennung <> 0:

\$FC, \$FD - Adresse der Fehler-  
          tabelle (LSB, MSB) (s.  
          u.)

Mit dieser Funktion wird das in \$FA angegebene Laufwerk selektiert. Zusätzlich wird in \$FC, \$FD die vor der letzten Deselektierung zuletzt angesprochene Spur uebergeben. Bei der ersten Selektierung ist sie \$FFFF, dieser Wert kann zur Initialisierung verwendet werden. Der Rueckgabewert in \$FB muss normalerweise auf 0 gesetzt werden, es sei denn bei der Initialisierung tritt ein Fehler auf. In diesem Fall wird dort die zugehoerige Fehlernummer und in \$FC, \$FD die Adresse der zum Treiber gehoerenden Fehlertabelle. Ein Laufwerk wird vor dem Aufruf einer anderen Funktion zunaechst selektiert!

3. Positionieren

Eingangsparameter: \$FA - Slot \* 16 + Laufwerk  
\$FC, \$FD - Spurnummer

Ausgangsparameter: keine

Der Treiber soll das in \$FA angegebene Laufwerk auf Spur \$FC, \$FD (LSB, MSB) positionieren. Evtl. auftretende Fehler muessen beim Schreiben, Lesen oder Formatieren gemeldet werden.

4. Lesen eines 256 Byte Sektors

Eingangsparameter: \$FA - Slot \* 16 + Laufwerk  
 \$FC, \$FD - Spurnummer  
 \$FE - Sektornummer  
 \$FF - MSB (Datenpuffer im AP 22 Speicher)

Ausgangsparameter: \$FB - Fehlerkennung (0=kein Fehler)  
 Status - Carry Clear = Fehler

falls Fehlerkennung <> 0:

\$FC, \$FD - Adresse der Fehler-tabelle (LSB, MSB) (s. u.)

Bei diesem Aufruf soll ein 256 Byte grosser Sektor von der Diskette in den Datenpuffer der AP 22 uebertragen werden. Das Laufwerk und der Slot des Disk Controllers sind in \$FA, die Nummer der anzusprechenden Spur, auf die zuvor bereits positioniert worden ist, in \$FC und \$FD, der zu lesende Sektor in \$FE und das hoeherwertige Byte des Datenpuffers im AP 22 Speicher in \$FF, angegeben. Ist das hoeherwertige Byte des Datenpuffers z. B. 0FEH so wird der Sektor ab Adresse 0FE00H bis Adresse 0FEFF in den AP 22 Speicher geschrieben. Tritt beim Lesen des Sektors von der Diskette ein Fehler auf so kann dieser als Fehlerkennung in \$FB zurueckgegeben werden. Zu dieser Kennung gehoert auch eine Fehlertabelle die am Ende des Abschnitts beschrieben ist.

5. Schreiben eines 256 Byte Sektors

Eingangsparameter: \$FA - Slot \* 16 + Laufwerk  
 \$FC, \$FD - Spurnummer  
 \$FE - Sektornummer  
 \$FF - MSB (Datenpuffer im AP 22 Speicher)

Ausgangsparameter: \$FB - Fehlerkennung (0=kein Fehler)  
 Status - Carry Clear = Fehler

falls Fehlerkennung <> 0:

\$FC, \$FD - Adresse der Fehler-tabelle (LSB, MSB) (s. u.)

Bei diesem Aufruf soll ein 256 Byte grosser Sektor aus dem Datenpuffer im AP 22 Speicher auf die angegebene Diskette geschrieben werden. Das Laufwerk und der Slot des Disk Controllers sind in \$FA, die Nummer der anzusprechenden Spur, auf die zuvor bereits positioniert worden ist, in \$FC und \$FD, die Nummer des Sektors, in den geschrieben werden soll, in \$FE und das hoeherwertige

Byte des Datenpuffers im AP 22 Speicher in \$FF, angegeben. Ist das hoehwertige Byte des Datenpuffers z. B. DFEH so werden in den Sektor die Daten der Adressen OFE00H bis OFEFFFH des AP 22 Speichers geschrieben. Tritt dabei ein Fehler auf so kann dieser als Fehlerkennung in \$FB zurueckgegeben werden. Zu dieser Kennung gehoert auch eine Fehlertabelle die am Ende des Abschnitts beschrieben ist.

#### 6. Formatieren einer Spur

Eingangsparameter: \$FA - Slot \* 16 + Laufwerk  
\$FC, \$FD - Spurnummer  
\$FF - MSB (Datenpuffer im AP 22 Speicher)

Ausgangsparameter: \$FB - Fehlerkennung (0=kain Fehler)  
Status - Carry Clear = Fehler

falls Fehlerkennung < 0:

\$FC, \$FD - Adresse der Fehler-  
tabelle (LSB, MSB) (s.  
u.)

Bei diesem Aufruf soll die in \$FC und \$FD angegebene Spur der, durch den Wert in \$FA, angegebenen Diskette formatiert werden. Auf die Spur ist zuvor bereits positioniert worden. Der Datenbereich der Formatierung soll mit den Daten im Datenpuffer der AP 22 gefuellt werden (dieser enthaelt 256 Byte \$E5). Ein evtl. auftretender Fehler kann als Fehlerkennung zurueckgegeben werden. Die dazugehoerende Tabelle wird anschliessend beschrieben. Dieser Aufruf wird nur vom Disk Manager Programm benutzt und kann, wenn ein anderes Formatierprogramm verwendet wird, weggelassen werden.

BIOS Details

Abschnitt 4

Bei den meisten Disketten Operationen ist die Rueckgabe einer Fehlermeldung moeglich. Neben der Fehlerkennung, die einer Fehlernummer entspricht, muss zusaetzlich die Adresse einer Fehlertabelle (im Apple IC Speicher) angegeben werden. In dieser Tabelle werden der Fehlertext der zugehoerigen BIOS Fehlermeldung und die moeglichen Quittierungen (im Quittierungsbyte) fuer jeden Fehler einzeln festgelegt. Sie ist wie folgt aufgebaut:

- Byte 0: Anzahl der definierten Fehler
- Byte 1, 2: Adresse des Fehlertextes fuer Kennung 1 (LSB, MSB)
- ...
- Byte x-1, x: Adresse des Fehlertextes fuer die Letzte Kennung (entspricht der Nummer in Byte 0)

Der Fehlertext besteht aus einer ASCII Zeichenfolge mit Bit 7 = 0. Er sollte nicht laenger als 15 Zeichen sein und wird mit dem Quittierungsbyte abgeschlossen. In diesem Byte stehen die 5 niederwertigen Bits fuer je eine Quittierungsmoeglichkeit, Bit 7 ist immer gesetzt. Ist eines der Bits gesetzt so ist die zugehoerige Option zulaessig, andernfalls nicht. Die Bits stehen fuer die folgenden Optionen:

- Bit 0: Option N = Wiederholung ist zulaessig
- Bit 1: Option W = Der Fehler darf weitergegeben werden
- Bit 2: Option B = Abbruch durch Warmboot ist zulaessig
- Bit 3: Option I = Ignorieren des Fehlers ist zulaessig
- Bit 4: Option K = Abbruch durch "Kaltstart" ist zulaessig
- Bit 5, 6: immer 0
- Bit 7: immer 1

Als Beispiel hier die Fehlertabelle des Apple Disk Treibers:

```

;*****
;
; FEHLERMELDUNGEN DES DISK IC TREIBERS
;
; Konstanten fuer Quittierungsbyte
;
0001 = OPTN EQU 1 ;Option N zulaessig
0002 = OPTW EQU 2 ;Option W zulaessig
0004 = OPTB EQU 4 ;Option B zulaessig
0008 = OPTI EQU 8 ;Option I zulaessig
0010 = OPTK EQU 10H ;Option K zulaessig
;
0080 = BASE EQU 80H ;Grundwertfuer Q.-byte
;
;.....
;
0000 = GERMAN EQU 0
FFFF = ENGLISH EQU NOT 0
;
0000 = LANGUAGE EQU GERMAN
;
;-----
;
; Anfang der Fehlermeldungstabelle
;
0000 05 ADRTAB: DB 5 ;Anzahl der Fehler
0001 0800 DW DSK01 ;Fehler 1
0003 1400 DW DSK02 ;Fehler 2
0005 1800 DW DSK03 ;Fehler 3
0007 2700 DW DSK04 ;Fehler 4
0009 2000 DW DSK05 ;Fehler 5
;
; Fehler 1: Hardware Fehler
;
000B 4861726477DSK01: DB 'Hardware'
0013 9F DB OPTN+OPTW+OPTB+OPTI+OPTK+BASE ;Quittie.
;
; Fehler 2: Defekte Formatierung
;
0014 466F726061DSK02: DB 'Format'
001A 9F DB OPTN+OPTW+OPTB+OPTI+OPTK+BASE ;Quittier.
;
; Fehler 3: Positionieren nicht moeglich
;
; DSK03:
001B 506F736974 IF LANGUAGE XOR NOT GERMAN
0026 9F DB 'Positionier'
DB OPTN+OPTW+OPTB+OPTI+OPTK+BASE ;Quittier.
ENDIF
;
;
IF LANGUAGE XOR NOT ENGLISH
DB 'Seek'
DB OPTN+OPTW+OPTB+OPTI+OPTK+BASE ;Quittie.
ENDIF

```

```

;
; Fehler 4: Daten Pruefsumme nicht korrekt
;
DSK04:
IF LANGUAGE XOR NOT GERMAN
DB 'Daten'
DB OPTN+OPTW+OPTB+OPTI+OPTK+BASE ;Quittier.
ENDIF
;
IF LANGUAGE XOR NOT ENGLISH
DB 'Data'
DB OPTN+OPTW+OPTB+OPTI+OPTK+BASE ;Quittier.
ENDIF
;
; Fehler 5: Diskette ist schreibgeschuetzt
;
DSK05:
IF LANGUAGE XOR NOT GERMAN
DB 'Schreibschutz'
DB OPTN+OPTW+OPTB+OPTI+OPTK+BASE ;Quittier.
ENDIF
;
IF LANGUAGE XOR NOT ENGLISH
DB 'Write protected'
DB OPTN+OPTW+OPTB+OPTI+OPTK+BASE ;Quittier.
ENDIF
;
003B END

```

Wie kann ein neuer Disk Treiber ins System integriert werden?

Da auf den Boot Spuren der Apple Diskette nahezu kein Platz fuer weitere Treiber vorhanden ist muss ein solcher Treiber durch ein Installationsprogramm eingefuegt werden. Es gibt nun 2 Moeglichkeiten: 1. Integration des Disk Treibers ohne BIOS Unterstuetzung und 2. mit BIOS Unterstuetzung.

Wird von der 1. Moeglichkeit Gebrauch gemacht, so muss das Installationsprogramm den Treiber an eine freie Position im Apple Speicher transferieren, der Bereich \$1C56 bis \$1FFF steht da fuer zur Verfuegung, die Einsprungadressen in der Treiber Adress Tabelle eintragen, den DPB einbauen, die notwendigen Adressen im DPH setzen und dabei die Bereiche fuer ALV und CSV belegen, und Basis Treibernummer und Wiederholungsanzahl in der Disk Konfigurationstabelle eintragen. Werden alle angegebenen Funktionen erfuehlt, so braucht im BIOS auf der Boot Spur nichts geändert werden. Der Disk Controller wird in diesem Fall waehrend der Kaltstart Phase nicht erkannt und somit dort auch nicht ins System eingebaut, dies geschieht dann durch das Installationsprogramm. Eine automatische Erkennung des Disk Controllers ist in diesem Fall nicht moeglich.

Im 2. Fall muessen die in diesem Abschnitt angegebenen Tabellen fuer den Disk Controller angepasst und an den entsprechenden Stellen in die Datei IBSCPM64.COM eingebaut werden. Bis auf den eigentlichen Treiber und dessen Einsprungadressen muss alles vorhanden sein, auch der DPB. Treiber und Einsprungadressen werden nach dem Systemstart durch das Installationsprogramm eingetragen. Bitte beachten Sie: Das Boot Laufwerk und Laufwerk A: muessen Apple ][ Disk Laufwerke bleiben!

## 5. Hardware Beschreibung

In diesem Abschnitt sollen Ihnen einige Informationen ueber Aufbau und Arbeitsweise der AP 22 vermittelt werden. Zunaechst wird die Hardware anhand des Blockschaltbildes erlaeuert, anschliessend wird erklart wie die einzelnen Funktionen angesprochen werden koennen. Bezeichnungen die sie auch im Blockschaltbild finden sind fett gedruckt. Adressen die sich auf den Apple IC Speicher beziehen werden mit einem fuehrenden S-Zeichen angegeben (z. B. \$C080); die Adressen des 64 K x 8 RAM's auf der AP 22 mit einem anschliessenden H (z. B. OFF00H). Alle Adressen sind als Hexadezimalzahl angegeben.

### 5.1 Aufbau und Funktionsweise

Im Blockschaltbild finden Sie oben den Apple Slot. Von diesen aus gesehen belegt die AP 22 die, fuer jeden Slot vorhandenen, Bereiche "Peripheral Card ROM Space" und "Peripheral Card I/O Space".

#### 5.1.1 Der "Peripheral Card I/O Space" Bereich

Der "Peripheral Card I/O Space" Bereich ist 16 Byte gross und wird mit dem Signal Dev. Select angesprochen, er belegt die Apple IC Adressen \$C0n0 bis \$C0nF, wobei n fuer die Slot-Nr. + 8 steht (8 bis \$F). Ueber diese Adressen werden mehrere Funktionen ausgefuehrt:

1. 8 oder 12K Byte grosse Bereiche des 64Kx8 RAM der AP 22 koennen in den Adressbereich \$0000 bis \$FFFF des Apple IC, anstelle des Autostart ("F8") ROM's (Adr. \$F800 bis \$FFFF) und der Applesoft ROM's (Adr. \$0000 bis \$F7FF), eingeblendet werden. Ist dies der Fall, so steht der Apple Monitor des "F8"-ROM's nicht mehr zur Verfuegung, d. h. z. B. Programme duerfen keine Unterbrechungspunkte (BRK) enthalten waehrend sie einen AP 22 Speicherbereich selektiert haben! Das Einblenden von 8 oder 12 K Byte grossen Teilbereichen ermöglicht einen schnellen Datentransfer zwischen dem AP 22 und dem Apple Speicher. Insbesondere ist es auch moeglich Programme fuer die Z80 CPU zu uebertragen ohne ein Programm auf dieser CPU laufen lassen zu muessen.
2. Zusammen mit der Speicher Selektierung wird die Z80 B CPU angehalten oder gestartet. Zu diesem Zweck wird das BUSRQ Signal verwendet, das den Z80 veranlasst den AP 22 internen Bus freizugeben und damit fuer den Apple zuganglich zu machen. Der Zustand des BUSRQ Signals wird durch die rote Leuchtdiode angezeigt; ist die LED eingeschaltet, so arbeitet der Z80, andernfalls kann der interne Bus vom Apple benutzt werden.
3. Unabhaengig davon kann ein von der AP 22 generiertes IRQ-Signal zurueckgesetzt werden oder aber ein Interrupt fuer den Z80 (INT-Signal) erzeugt werden.

Wie dies im einzelnen geschieht finden Sie im Abschnitt 5.2. Die Daten, die sich beim Schreiben oder Lesen dieser Adressen auf dem Datenbus befinden, werden von der AP 22 nicht verwendet.

### 5.1.2 Der "Peripheral Card ROM Space" Bereich

Der 256 Byte grosse "Peripheral Card ROM Space" Bereich wird mit dem Signal I/O Select angesprochen und belegt die Apple IC Adressen \$Cn00 bis \$CnFF, wobei n fuer die Slot-Nr. steht (1 bis 7). Dieser Bereich wird auf einen 256 Byte grossen Teilbereich des 64K x 8 RAM der AP 22 abgebildet. Die oberen 8 Adressbits dieses Teilbereichs werden durch 8 Widerstaende festgelegt, die Voreinstellung ist 0FFF0H bis 0FFFFH. Die Widerstaende finden Sie auf der Bestueckungsseite der AP 22 zwischen Slot Stecker und 64 K RAM Bereich. Es sind entweder 8 Einzelwiderstaende zu je 1 K Ohm oder aber 1 Widerstandsnetzwerk 8 x 1 kOhm im 16 poligen DIL-Gehaeuse fuer die I/O Bereichs Selektierung installiert. Die dem Slot Stecker zugewandten 8 Loetpunkte stehen fuer die 8 hoehwertigen Bits der Adresse und koennen ueber Bruecken auf +5V oder 0V gelegt werden (Voreinstellung: alle auf +5V). Der der Apple Rueckwand am naechsten liegende Widerstand steht fuer Bit 15 der Adresse, der davorliegende fuer Bit 14, ..., der am weitesten zur Apple Tastatur liegende fuer Bit 8 (ab Leiterplattenversion 4.1, s. u.). Liegt der jeweilige Loetpunkt auf +5V so wird eine 1, liegt er auf 0V eine 0 fuer das zugehoerige Bit angenommen. Hier noch einmal die Belegung:

Ab Leiterplattenversion V4.1:

<- Apple Tastatur 8 9 10 11 12 13 14 15 Apple Rueckwand ->

In der Leiterplattenversion V4 ist die Belegung wie folgt:

<- Apple Tastatur 9 8 11 10 13 12 15 14 Apple Rueckwand ->

Der 256 Byte Bereich wird uebrigens im AP 22 BIOS nicht benutzt um den Betrieb der AP 22 im Slot 0 zu ermoeeglichen.

Bedingt durch die asynchrone Arbeitsweise der AP 22 und der Erzeugung des I/O Select Signals durch den Apple IC, ist der Zugriff auf diesen Bereich nur bei angehaltener Z80 B CPU moeglich. Bei laufender Z80 B CPU wird beim Schreiben auf eine beliebige Adresse dieses Bereichs ein nicht maskierbarer Interrupt (NMI) fuer den Z80 erzeugt.

5.1.3 Arbeitsweise

Die Z80 B CPU stellt zusammen mit dem 64K x 8 RAM und dem 12 MHz Quarzoszillator einen eigenstaendigen Mikrocomputer dar, der vom Apple lediglich gesteuert werden kann. Dadurch kann die Z80 B CPU mit ihrer vollen Taktfrequenz von 6 MHz, ohne irgendwelche Wartezyklen, betrieben werden. Da die AP 22 mit einer eigenstaendigen Taktversorgung arbeitet, die nicht synchron zum Apple Takt ist, spricht man von einer asynchronen Arbeitsweise. Dadurch bedingt koennen Signale vom Apple erst nach einer Synchronisierung auf den Takt der AP 22 benutzt werden, dies geschieht im Funktionsblock Synchronisation. Die Steuersignale fuer das 64 K x 8 RAM, die Adressmultiplexer und den DatenBuffer zwischen Apple und AP 22 sowie das Reset-Signal fuer den Z80 werden in der Zentralen Zeitsteuerung, in Abhaengigkeit von den synchronisierten Apple JC und Z80 Signalen, erzeugt. Diese arbeitet synchron zum 12 MHz Quarzoszillator und verwendet nur synchronisierte Apple Signale. Das Refresh Signal fuer die Speicher wird aus dem Apple /01-Signal dividiert durch 8 erzeugt, die RAM Bausteine arbeiten mit eingebautem "Self Refresh".

Beim Versuch sowohl vom Z80 wie auch vom Apple JC gleichzeitig auf den 64 K x 8 RAM Bereich zuzugreifen, wird zunaechst der laufende Z80 Speicherzyklus komplett abgearbeitet, erst dann wird der Apple Speicherzugriff ausgefuehrt. Hierzu ist es erforderlich, den 6502 solange ueber das RDY-Signal zu verzoegern, bis die Daten konstant anliegen (i. d. R. 1 Zyklus). Da das RDY-Signal nur bei Lesezyklen vom 6502 ausgewertet wird, ist ein schreibender Zugriff bei laufendem Z80 normalerweise nicht moeglich, es sei denn er folgt unmittelbar einem Lesezyklus. Dies ist nur der Fall wenn der Schreibbefehl im AP 22 Speicher selbst ausgefuehrt wird. Der Z80 laeuft nach Beendigung des 6502 Speicherzyklusses weiter.

Zusammengefasst die Speicherzugriffsmoeglichkeiten:

bei angehaltenem Z80:

Speicher selekt.	6502 lesend	6502 schreibend	I/O Bereich
Nein	Nein	Nein	Ja
Ja	Ja	Ja	Ja

bei laufenden Z80: (\*) = nicht auf dem ITT 2020

Nein	Nein	Nein	Nein
Ja	Ja (*)	Nein	Nein



#### 5.1.4 Interrupts und Reset

Die Z80 B CPU kann durch einen HALT oder einen IN bzw. OUT Befehl auf die Z80 I/O Ports 0 bis 7FH einen IRQ fuer den 6502 des Apple IC erzeugen. Dieser IRQ muss vom 6502 durch ansprechen der Adresse \$C0n0 (mit n = Slot-Nr. + 8, Slot-Nr. 0 bis 7) quittiert werden, bis dahin bleibt das IRQ-Signal auf 0-Potential.

Ein maskierbarer Interrupt fuer den Z80 (INT-Signal) kann, wenn Jumper J3 offen ist, vom 6502 durch Ansprechen von Adresse \$C0n8 (n siehe oben) erzeugt werden, ist J3 geschlossen (Voreinstellung) so werden diese in Abständen von 10 ms durch den 100 Hz Zeitgeber erzeugt. Diese Interrupts werden durch den Interrupt Acknowledge Zyklus der Z80 CPU quittiert.

Durch das Apple Reset Signal wird der AP 22 Speicher fuer den Apple deselektiert und ein evtl. anliegender IRQ zurueckgenommen. Ausserdem wird, wenn Jumper J1 geschlossen und Jumper J2 offen ist, die Z80 B CPU zurueckgesetzt. Ist J1 offen und J2 geschlossen, so wird fuer die Z80 CPU nur ein Power on Reset erzeugt.

Die Zeitbasis fuer den 100 Hz Zeitgeber ist der Apple/01 Takt. Dadurch ist der 100 Hz Takt von der zentralen Apple Zeitbasis abhaengig. Diese arbeitet leider mit 2 verschiedenen Taktfrequenzen, auf manchen Rechnern mit 14,31818 MHz, auf anderen mit 14,25 MHz. Wird der Zeitgeber fuer eine Uhr eingesetzt, so muessen die folgenden Abweichungen beruecksichtigt werden:

Zeitraum	Abweichung 14,31818 MHz	Abweichung 14,25 MHz
1 Sekunde	0,0012445 Sekunden	0,006035 Sekunden
1 Minute	0,0746 Sekunden	0,362 Sekunden
1 Stunde	4,48 Sekunden	21,73 Sekunden
1 Tag	107,5 Sekunden	521,4 Sekunden

Tabelle 9: mittlere Zeitabweichungen des 100 Hz Zeitgebers

#### 5.1.5 Betriebsbedingungen

Stromverbrauch: 5 V / 750 bis 1000 mA  
 Umgebungstemperatur: 15 bis 35 Grad C

Die AP 22 kann in einem der Slots 0 bis 6 des Apple IC betrieben werden. Bitte beachten Sie dass die 5 V Versorgungsspannung des Apple IC Netzteils bei gemeinsamen Betrieb der AP 22 und anderer Peripherikarten mit hohem Stromverbrauch, ueberlastet werden kann. In diesen Faellen ist es empfehlenswert ein staerkeres Netzteil einzusetzen, um einen zuverlaessigen Betrieb zu gewaehrleisten.

5.2 Ansprache der AP 22 durch den Apple II

Welche Bedeutung haben die Adressen des "Peripheral Card I/O Space" Bereichs ?

Nachfolgend steht n innerhalb einer Adresse fuer Slot-Nr. + 8.

Die Adressen \$C0n1 bis \$C0n7 selektieren bzw. deselektieren einen 12K oder 3K Byte grossen Speicherbereich des 64 K x 8 RAM und halten die Z80 B CPU an, die Adressen \$C0n9 bis \$C0nF selektieren die gleichen Speicherbereiche und starten den Z80. Ein Lesezyklus auf eine dieser Adressen bewirkt einen zusatzlichen Schreibschutz des gesamten Speichers, in das 64 K x 8 RAM kann nur nach einer Selektierung bzw. Deselektierung mit einem Schreibbefehl geschrieben werden. Der Schreibschutz wirkt auch auf den "Peripheral Card I/O ROM Space" Bereich ! Welche Bereiche des AP 22 RAM's jeweils selektiert sind finden Sie in der auf der naechsten Seite folgenden Tabelle.

Die verbleibenden Adressen haben die folgende Bedeutung:

- \$C0n0: Lesebefehl - Quittierung eines IRQ's  
 Schreibbefehl - dgl. und Ruecknehmen des Z80 INT-Signals
- \$C0n8: Lesebefehl - Quittierung eines IRQ's  
 Schreibbefehl - dgl. und Erzeugung eines maskierbaren Interrupts (INT-Signal) fuer den Z80 (nur wenn J3 offen ist)

Adr. SC0nX	Apple ][ Adressbereich	AP 22 RAM Adressbereich
X=1 oder 9	AP 22 Speicher abgeschaltet. (deselektiert)	
X=2 oder \$A	\$E000 bis \$EFFF	0000H bis 0FFFH
X=4 oder \$C	\$0000 bis \$0FFF	1000H bis 1FFFH
X=4 oder \$C	\$E000 bis \$EFFF	2000H bis 2FFFH
X=4 oder \$C	\$F000 bis \$FFFF	3000H bis 3FFFH
X=2 oder \$A	\$0000 bis \$0FFF	4000H bis 4FFFH
X=6 oder \$E	\$0000 bis \$0FFF	5000H bis 5FFFH
X=6 oder \$E	\$E000 bis \$EFFF	6000H bis 6FFFH
X=6 oder \$E	\$F000 bis \$FFFF	7000H bis 7FFFH
X=3 oder \$B	\$E000 bis \$EFFF	8000H bis 8FFFH
X=5 oder \$D	\$0000 bis \$0FFF	9000H bis 9FFFH
X=5 oder \$D	\$E000 bis \$EFFF	0A000H bis 0AFFFH
X=5 oder \$D	\$F000 bis \$FFFF	0B000H bis 0BFFFH
X=3 oder \$B	\$0000 bis \$0FFF	0C000H bis 0CFFFH
X=7 oder \$F	\$0000 bis \$0FFF	0D000H bis 0DFFFH
X=7 oder \$F	\$E000 bis \$EFFF	0E000H bis 0EFFFH
X=7 oder \$F	\$F000 bis \$FFFF	0F000H bis 0FFFFH

Tabelle 10: AP 22 Speicherelektierung

Die Tabelle ist folgendermassen zu lesen: Wird die Adresse SC0nX angesprochen, so liegt im Apple ][ Adressbereich \$000 bis \$FFF der Adressbereich .000H bis .FFFH des AP 22 64 K x 8 RAM's. Wird der Bereich mit dem kleineren X-Wert selektiert, so wird der 280 angehalten, wird der grössere Wert verwendet, gestartet. Wird der Bereich durch einen Schreibbefehl selektiert, so ist Schreiben in diesen Bereich und in den "Peripheral Card I/O ROM Space" Bereich möglich, wird ein Lesebefehl verwendet, so koennen diese Bereiche auch nur gelesen werden.

## Beispiel 1:

In die Speicherzelle 5555H der AP 22 soll der Wert \$5A geschrieben werden. Die AP 22 steckt in Slot 4.

```
STASC0C6 ;Speicherbereich selektieren, Z80 stoppen
LDA #55A
STA $D555 ;Wert eintragen
LDA SCOC1 ;Speicher deselekt., Schreibschutz, Z80 Stop
```

## Beispiel 2:

Die Speicherzelle DAAAAH der AP 22 soll gelesen werden, der Z80 soll angehalten und der Speicher schreibgeschuetzt werden. Die AP 22 soll in Slot 2 stecken.

```
LDA $COA5 ;RAM selektieren, Z80 stoppen, Schreibschutz
LDA $EAAA ;Speicherzelle lesen
PHA ;und retten
LDA $COA1 ;RAM deselekt., Z80 stoppen, Schreibschutz
PLA
```

## Beispiel 3:

In die Speicherzelle 123H der AP 22 soll der Wert \$45 eingetragen werden. Das RAM soll nachher schreibgeschuetzt sein und der Z80 laufen. Die AP 22 soll in Slot 5 stecken.

```
STA $C0D2 ;RAM selektieren, Z80 stoppen
LDA $45
STA $E123 ;Wert eintragen
LDA $C0D9 ;RAM deselekt., Z80 starten, Schreibschutz
```

6. Kompatibilität zum SoftCard CP/M

In diesem Abschnitt sollen einige Unterschiede zwischen dem IBS Apple II CP/M und dem SoftCard CP/M aufgezeigt werden. Diese Unterschiede ergeben sich zum einen aus der grösseren TPA, dadurch bedingt liegen einige spezielle Bereiche des SoftCard CP/M jetzt innerhalb des BDOS, und zum anderen aus den getrennten Speicherbereichen. Die Anpassungen fuer MBASIC und GBASIC sind hier ebenfalls beschrieben. Neben den hier aufgeführten Unterschieden kann es durchaus auch noch andere geben - in diesem Abschnitt sind nur die wichtigsten Punkte aufgefuehrt.

6.1 Allgemeines

Die Unterschiede zwischen SoftCard CP/M und AP 22 CP/M sind sowohl durch die Hardware wie auch durch das jeweilige BIOS bedingt. Hardwarebedingte Unterschiede sind:

## 1. Umschaltung zwischen Z80 und 6502 Modus

Die SoftCard kann durch Schreiben in Ihren "Peripheral Card I/O ROM" - Bereich zwischen Z80 und 6502 Programmen umschalten. Dies ist mit der AP 22 natuerlich nicht mehr moeglich. Hier arbeiten 6502 und Z80 weitgehend parallel. Der Aufruf von 6502 Unterprogrammen kann ueber Restart Befehle erfolgen (RST 1 und RST 4/4).

## 2. Gemeinsamer Speicher fuer Z80 und 6502

Im SoftCard System wird sowohl fuer den Z80 wie auch fuer den 6502 des Apple der Apple Speicher benutzt. Dadurch war es moeglich 6502 Programme durch den Z80 direkt zu laden. Dies ist im AP 22 System nicht moeglich, hier koennen aber 6502 Programmteile ueber die Restart Befehle in den Apple Speicher transferiert werden (RST 4/0 und RST 4/1). Fuer Benutzerprogramme sollte der Speicherbereich \$200 bis \$3EF des Apple Speichers verwendet werden.

Nun zu den Software bedingten Unterschieden zwischen SoftCard BIOS und AP 22 BIOS:

1. Die "Hardware/Software Screen Function Table"

In dieser Tabelle werden im SoftCard BIOS die Steuerzeichen fuer die Video Ein-/Ausgabe definiert. Diese Tabelle ist im AP 22 BIOS, allerdings an anderer Stelle, ebenfalls vorhanden. Sie befindet sich in der Page \$C des Apple Speichers, es kann allerdings maximal fuer jeden Slot eine solche Tabelle definiert sein, im gelieferten System ist aber nur eine enthalten.

2. Die "Keyboard Character Redefinition Table"

Mit dieser Tabelle codiert das SoftCard BIOS einige Eingabezeichen um. Im AP 22 BIOS ist zu diesem Zweck ein recht umfangreiches Eingabe Umcodier System implementiert worden in dem, neben einfachen Umcodierungen, auf einzelne Tasten oder Tastenfolgen laengere Texte gelegt werden koennen, wie z. B. auf die Tastenfolge <ESC> M der Text MBASIC. Mit diesem System koennen somit auch "Funktionstasten" erzeugt werden.

3. Bereiche fuer "User I/O driver software"

Im SoftCard BIOS sind 3 128 Byte grosse Bereiche fuer die Installation anderer Treiber vorgesehen. Diese sind im AP 22 System nicht mehr enthalten, sie wuerden sich dort innerhalb des BIOS befinden. Im AP 22 Speicher ist nur der Speicherbereich unterhalb des BIOS Stack frei (s. Speicherbelegung), im 6502 Teil ist der Bereich \$1C56 bis \$1FFF fuer neue Treiber vorgesehen.

4. Die "I/O Vector Table"

In dieser Tabelle koennen im SoftCard BIOS die Einsprungsadressen fuer die Treiber der Text Ein-/Ausgabe Peripheriekarten veraendert werden. Sie ist im AP 22 BIOS nicht enthalten, fuer diesen Zweck kann dort aber die Zuordnungstabelle logische -> physikalische CP/M Geraete in weit flexiblerer Form benutzt werden.

5. Die "Card Type Table"

Mit dieser Tabelle kann man im SoftCard CP/M feststellen welche Peripheriekarte in welchem Slot installiert ist. Diese Tabelle ist im AP 22 BIOS ebenfalls, allerdings an anderer Stelle, in aehnlicher Form enthalten. Sie finden sie ab Adresse OFF11H bis OFF17H im AP 22 Speicher.

### 6. Sektor skewing bei Apple Disketten

Um eine vertretbare Geschwindigkeit beim Lesen von Apple Disketten zu erreichen, wird beim sequentiellen Zugriff durch das BIOS jeweils jeder 3. Sektor gelesen (Skew Faktor 3). Im SoftCard BIOS wird diese Sektornummer Umsetzung im Apple Disk Treiber vorgenommen, im AP 22 BIOS wird, im Gegensatz dazu, die XLT Tabelle des BIOS zu diesem Zweck verwendet. Der beim BIOS Aufruf SETSEC angegebene Sektor wird im AP 22 BIOS somit direkt angesprochen und nicht mehr umgesetzt; ist eine Umsetzung erforderlich so muss zuvor SECTRAN aufgerufen werden.

6.2 MBASIC und GBASIC

Fuer die Microsoft Basic Interpreter MBASIC.COM und GBASIC.COM sind bereits die notwendigen Anpassungen vorhanden. Sie werden waehren der Installationsphase in die Originalprogramme eingebaut. Die Anpassung erwartet eine evtl. vorhandene 80 Zeichen Karte in Slot 3, die Anfangsadresse der zugehoerigen Ausgabe Uncodier Tabelle darf nicht veraendert worden sein; ausserdem ist der Apple Speicher von \$200 bis \$27F mit 6502 Unterprogrammen belegt. Bei manchen BASIC-Befehlen ergeben sich Unterschiede zur SoftCard Version. Diese werden hier einzeln beschrieben.

Unterschiede in MBASIC und GBASIC:

## 1. GR-Befehl

Bei Verwendung einer 80 Zeichen Karte wird der Cursor auf Zeile 4 positioniert und nicht auf 24.

## 2. POKE-Befehl

Mit dem POKE-Befehl koennen Daten in den AP 22 Speicher geschrieben werden. Das angegebene Byte wird in die angegebene AP 22 Speicherzelle geschrieben. Der 6502 Speicher ist auf diese Weise nicht zugaenglich. Ein "POKE" in den Apple Speicher kann mit:

```
POKE65=555
CALL %POKE65 (<Adr. LSB>,<Adr. MSB>,<Wert>)
```

erfolgen. Adr. ist die gewuenschte Apple Adresse in die der <Wert> geschrieben werden soll.

## 3. WAIT

Die im WAIT-Befehl angegebene Adresse wird als Apple IC Speicheradresse aufgefasst.

Beispiel:

```
100 WAIT %HC000,128
200 PRINT "Tastendruck!":GOTO 100
```

## 4. PEEK

Die im PEEK-Befehl angegebene Adresse wird als AP 22 Adresse aufgefasst. Ein Zugriff auf Apple IC Speicherzellen ist nicht moeglich.

Leichte Abweichungen koennen sich auch beim BEEP, BUTTON und PDL Befehl ergeben.

## Unterschiede im GBASIC:

Die High Resolution Grafik (HGR) des GBASIC wird in dieser Anpassung zunaechst im Bereich 1000H bis 2FFFF des AP 22 Speichers aufgebaut. Der Grund hierfuer ist der, das im SoftCard GBASIC alle HGR-Befehle von der Z80 CPU ausgefuehrt werden, der angegebene Adressbereich war dort gleichzeitig der HGR1-Bereich des Apple IIc. Dies lies sich im Binaercode des GBASIC leider nicht aendern. Um nun aber trotzdem die High Resolution Graffk benutzen zu koennen ist eine zusaetzliche Kopiermoeglichkeit, durch die der angegebene Bereich in den HGR1-Bereich des Apple transferiert wird, eingebaut worden. Die Befehle im einzelnen:

## 1. HGR

Der HGR-Befehl hat, neben den in der SoftCard beschriebenen, die zusaetzliche Funktion den Grafik-Bereich im AP 22 Speicher in den HGR1-Bereich des Apple zu kopieren. Nachdem eine Grafik erstellt worden ist kann diese z. B. mit HGR 3 auf dem Bildschirm sichtbar gemacht werden.

## 2. H PLOT

Die gezeichneten Linien werden im AP 22 Speicher abgelegt und erst nach einem HGR 2 oder HGR 3 Befehl auf dem Bildschirm sichtbar.

## 3. HSCRN

HSCRN bezieht sich auf das im AP 22 Speicher stehende und nicht auf das tatsaechlich dargestellte Bild.

## Beispiel:

```
10 HGR: REM Der HGR-Bereich wird geloescht
20 HCOLOR=2
30 H PLOT 24,125 TO 100,12 TO 270,1
40 HGR 2: REM Erst jetzt werden die Linien sichtbar!
```

Anhang ATreiber fuer 80 Zeichen Karte, serielles und paralleles Interface

Die hier abgedruckten Treiber werden fuer die Text Ein- / Ausgabe Peripheriekarten in den Apple II Slots benutzt. Welcher der 3 Treiber benutzt wird haengt von den Erkennungsbytes der Karte ab, d. h. z. B. das ein serielles Interface nicht statt mit dem seriellen Interface Treiber auch mit dem 80 Zeichen Treiber arbeiten kann. Als was die jeweilige Karte erkannt worden ist, steht in der Karten Typ Tabelle ab Adresse 8FF11H (fuer Slot 1) bis Adresse 8FF17H (fuer Slot 7) im AP 22 Speicher. Ist der dort angegebene Typ 3, so wird der serielle Interface, ist er 4, der 80 Zeichen, und ist er 5, der parallele Interface Treiber verwendet. Die im Listing angegebene Anfangsadresse kann sich in spaeteren Versionen aendern, entscheidend sind die Einsparungsadressen fuer die Treiber. Wo welche Adresse eingetragen ist wird in der folgenden Tabelle angegeben:

Apple II Adresse : Label im Listing : Bedeutung

\$048, \$049	: INIT80	: Initialisierung der 80 Zeichen Karte
\$04C, \$04D	: INP80	: Eingabe ueber 80 Zeichen Karte
\$04E, \$04F	: OUTP80	: Ausgabe ueber 80 Zeichen Karte
\$050, \$051	: SIINI	: Initialisierung serielles Interface
\$052, \$053	: SISTAT	: Eingabestatus serielles Interface
\$054, \$055	: SIIN	: Eingabe vom seriellen Interface
\$056, \$057	: SIOUT	: Ausgabe auf dem seriellen Interface
\$058, \$059	: PPINI	: Initialisierung paralleles Interface
\$05A, \$05B	: PPSTAT	: Ausgabe Status paralleles Interface
\$05C, \$05D	: PPIN	: "Eingabe" vom parallelen Interface (Dummy)
\$05E, \$05F	: PPOUT	: Ausgabe auf dem parallelen Interface

Die Adressen der angegebenen Label stehen in den zugehörigen Apple II Speicherzellen in der Reihenfolge niederwertiges, hoehwertiges Byte. Die Inhalte der Apple Speicherzellen finden Sie an gleicher Stelle in der Datei IBSCPM64.COM wenn Sie diese mit DDT ohne Offset laden. Geaenderte Treiber und Treiber Adressen muessen Sie in IBSCPM64.COM eintragen. Den Quellcode fuer die abgedruckten 3 Treiber finden Sie auf der "AP 22 - CP/M STARTER" - Diskette unter der Bezeichnung AP8010.S. Er kann mit dem Assembler/Editor BIG MAC veraendert und neu assembliert werden. Anschliessend muss der Object Code von der DOS 3.3 Diskette auf eine CP/M Diskette uebertragen werden, damit er in IBSCPM64.COM eingebunden werden kann. Dazu muss zunaechst der Binaercode in den Apple II Speicher ab Adresse \$4000 geladen werden. Beim anschliessenden Starten des AP 22 CP/M Systems bleibt dieser Bereich unveraendert, er wird erst belegt wenn die Pseudodisk (Pi) oder das Autoexecute Programm (AEXEC.COM) benutzt werden. Dieser Bereich kann dann durch ein kleines Programm, unter Ausnutzung des Restart 4 Speichertransfers, in den AP 22 Speicher uebertragen und dann mit SAVE gerettet werden. Durch Starten von IBSCPM64 kann nach den Aenderungen das modifizierte System auf die Diskette geschrieben werden.

Beispiel fuer den Speichertransfer:

Es wird angenommen das der Binaercode im Apple Speicher ab Adresse \$4000 bis \$40FF steht.

--- Zunaechst wird DDT aufgerufen

A>DDT

DDT Vers. 2.2

-4000

--- Das Transferprogramm wird ab 8000H eingegeben

8000 LXI H,4000 --- Startadresse im Apple Speicher

8003 LXI D,100 --- Zieladresse im AP 22 Speicher

8006 LXI B,100 --- Laenge des Binaercodes im Apple Speicher

8009 RST 4 --- Restart 4 Transfer

800A ^

-5000A --- Funktionsnummer eintragen

800A xx 01 --- xx steht fuer eine beliebige Hexzahl

800B xx ^ --- Ende der Eingabe

-0000,0000 --- Speichertransfer ausfuehren

\*000B

-00 --- Zurueck in den CCP

A>SAVE I TREIBER.COD

--- Der Binaercode des Treibers wird in eine CP/M Datei gerettet

Ihre Eingaben sind unterstrichen, mit --- beginnende Texte sind Kommentare.

```

1  *
2  *
3  *
4  * AP9810 - 80 Zeichen Treiber fuer Apple II CP/M auf AP22
5  *
6  * Version 1.0 vom 12. 5. 1983
7  *
8  * Teil des IBS AP22 BIOS
9  *
10 *
11 *
12 * Zero Page Adressen
13 *
14 CSWH EQU $37 ;MSB (CSW)
15 KSWH EQU $39 ;MSB (KSW)
16 *
17 * SoftCard Uebergabe Adressen
18 *
19 MAKKU EQU $45 ;AKku Uebergabe
20 MREGX EQU $46 ;Register X Uebergabe
21 MREGY EQU $47 ;Register Y Uebergabe
22 *
23 * Treiber-Parameter
24 *
25 PAR1L EQU $FA ;Slot-Nr.
26 PAR1H EQU $FB ;Slot-Nr.
27 PAR2H EQU $FD ;Eingabe Zeichen
28 PAR3L EQU $FE ;Ausgabe Zeichen
29 *
30 * ROM-Adressen
31 *
32 INI80Z EQU $C800 ;Initialisierung
33 IN80Z EQU $C840 ;Zeichen Eingabe
34 OUT80Z EQU $C9AA ;Zeichen Ausgabe
35 *
36 ROMOFF EQU $CFFF ;ROM deselektieren
37 *
38 * I/O Scratch Adressen
39 *
40 IOSCR4 EQU $678
41 IOSCR5 EQU $6FB
42 *
43 IOBASE EQU $C800

```

```

45 *
46 *
47 *
48 *   Die Anfangsadresse Kann sich in anderen Versionen aendern !
49 *
50       ORG $190B
51 *
52 *
53 *
54 *   INIT80 - Initialisierung der 80 Zeichen Karte
55 *
56 *   Dieses Unterprog. wird beim Kalt- und Warmstart aufgerufen
57 *
58 *   Eingangsparameter:
59 *
60 *   PAR1H - Slot-Nr.
61 *
62 *
63 *
190B: 20 07 1A 64  INIT80  JSR  ROMINI   ;ROM selektieren
190E: 86 37 65      STX  CSWH
190B: 86 39 66      STX  KSWH
19E2: A9 80 67      LDA  #0       ;ASCII NUL
19E4: 4C 80 C8 68      JMP  INI80Z   ;Karte initialisieren
69 *
70 *
71 *
72 *   INP80 - 80 Zeichen Eingabe Treiber
73 *
74 *   Eingangsparameter: PAR1H - Slot-Nr.
75 *
76 *   Ausgangsparameter: PAR2H - eingegebenes Zeichen
77 *
78 *
79 *
19E7: 20 07 1A 80  INP80  JSR  ROMINI   ;Eingabe vorbereiten
19EA: 20 40 C8 81      JSR  IN80Z   ;Zeichen einlesen
19ED: A6 FB 82      LDX  PAR1H   ;Slot-Nr.
19EF: 8D 78 86 83      LDA  IOSCR4,X ;eingelesenes Zeichen
19F2: 85 FD 84      STA  PAR2H
19F4: 60 85      RTS
86 *
87 *
88 *
89 *   OUTP80 - 1 Zeichen auf der 80 Zeichen Karte ausgeben.
90 *
91 *   Eingangsparameter: PAR1H - Slot-Nr.
92 *                   PAR3L - Ausgabe Zeichen
93 *
94 *
95 *
19F5: 20 07 1A 96  OUTP80 JSR  ROMINI   ;ROM selektieren
19F8: A4 FB 97      LDY  PAR1H   ;Slot-Nr.
19FA: A5 FE 98      LDA  PAR3L   ;Ausgabe Zeichen
19FC: 99 78 86 99      STA  IOSCR4,Y
19FF: 85 45 100     STA  MAKKU
1A01: AC FB 86 101     LDY  IOSCR5   ;Y-Wert erneuern
1A04: 4C AA C9 102     JMP  OUT80Z

```

```

184 *
185 *-----*
186 *
187 *   ROMINI - ROM Bereich selektieren
188 *
189 *-----*
190 *
1A07: 20 10 1A 111 ROMINI JSR SELROM ;ROM Bereich selektieren und
1A0A: 8C FB 06 112 STY IOSCR5 ;Y-Werte setzen
1A0D: 84 47 113 STY MREGY
1A0F: 68 114 RTS
115 *
116 *-----*
117 *
118 *   SELROM - Slot-abhaengiges ROM selektieren
119 *
120 *   Eingangparameter: PAR1H - Slot-Nr.
121 *
122 *   Ausgangparameter: X - #Cn mit n=Slot-Nr.
123 *                     Y - Slot * 16
124 *                     A - Slot * 16
125 *
126 *-----*
127 *
1A18: AD FF CF 128 SELROM LDA ROMOFF ;alle ROM's abschalten
1A13: A9 C8 129 LDA #8C8 ;MSB (I/O Bereich)
1A15: 85 FB 130 ORA PAR1H ;+ Slot-Nr.
1A17: 85 47 131 STA MREGY
1A19: AA 132 TAX ;X=#Cn
1A1A: A8 98 133 LDY #8
1A1C: 84 46 134 STY MREGX ;LSB=8
1A1E: 81 46 135 LDA (MREGX),Y ;ROM ansprechen
1A28: 86 46 136 STX MREGX
1A22: 8A 137 TXA
1A23: 8A 138 ASL ;Slot*16 berechnen
1A24: 8A 139 ASL
1A25: 8A 140 ASL
1A26: 8A 141 ASL
1A27: A8 142 TAY
1A28: 84 47 143 STY MREGY
1A2A: 68 144 RTS

```

```

146 *
147 *
148 *
149 *   Treiber fuer Serial Communication Card und
150 *   Parallel Printer Interface
151 *
152 *   Version 1.8 - 12. 5. 1983
153 *
154 *   Copyright (C) 1983 by Rainer Ellenbrake
155 *
156 *   Teil des IBS AP22 BIOS
157 *
158 *
159 *
160 *   SIINI - Initialisierung der seriellen Schnittstelle
161 *
162 *   Eingangsparameter: PAR1H - Slot-Nr.
163 *
164 *
165 *
166 *
167 *
168 *
169 *
170 *
171 *
172 *
173 *
174 *
175 *   SISTAT - Eingabe Status des seriellen Interface bestimmen
176 *
177 *   Eingangsparameter: PAR1H - Slot-Nr.
178 *
179 *   Ausgangsparameter: PAR2H - Status
180 *
181 *
182 *
183 *
184 *
185 *
186 *
187 *
188 *
189 *

```

```

1A2B: 28 66 1A 166 SIINI   JSR X16
1A2E: A9 83 167         LDA #3
1A30: 99 8E C8 168         STA IOBASE+14,Y
1A33: A9 15 169         LDA #B15
1A35: 99 8E C8 170         STA IOBASE+14,Y
1A38: 68 171         RTS

```

```

1A39: 28 66 1A 183 SISTAT  JSR X16
1A3C: 89 8E C8 184         LDA IOBASE+14,Y ;Status Port
1A3F: 4A 185         LSR
1A40: A9 88 186         LDA #8
1A42: E9 88 187         SBC #8
1A44: 85 FD 188         STA PAR2H ;Status bestimmen (0=nicht bereit)
1A46: 68 189         RTS

```

```

191 *
192 *
193 *
194 *   SIIN - Eingabe von der seriellen Schnittstelle
195 *
196 *   Eingangsparameter: PAR1H - Slot-Nr.
197 *
198 *   Ausgangsparameter: PAR2H - eingelesenes Zeichen
199 *
200 *
201 *
202 *
1A47: 20 66 1A 202 SIIN   JSR   X16
203 *
1A4A: 89 8E C8 204 WAIT   LDA   IOBASE+14,Y
1A4D: 4A         205 LSR
1A4E: 98 FA     206 SCC   WAIT
207 *
1A50: 89 8F C8 208 LDA   IOBASE+15,Y
1A53: 85 FD     209 STA   PAR2H
1A55: 68       210 RTS
211 *
212 *
213 *
214 *   SIOUT - Ausgabe auf serieller Schnittstelle
215 *
216 *   Eingangsparameter: PAR1H - Slot-Nr.
217 *   PAR3L - Ausgabe Zeichen
218 *
219 *
220 *
1A56: 20 66 1A 221 SIOUT  JSR   X16
222 *
1A59: 89 8E C8 223 WAIT2  LDA   IOBASE+14,Y ;Ausgabepuffer leer ?
1A5C: 29 82    224 AND   #2
1A5E: F8 F9    225 BEQ   WAIT2
226 *
1A60: A5 FE    227 LDA   PAR3L ;Ausgabe Zeichen
1A62: 99 8F C8 228 STA   IOBASE+15,Y
1A65: 68       229 RTS
230 *
231 *
232 *
233 *   X16
234 *
235 *   Eingangsparameter: PAR1H - Slot-Nr.
236 *
237 *   Ausgangsparameter: Y - Slot-Nr. * 16
238 *
239 *
240 *
1A66: A5 FB    241 X16   LDA   PAR1H ;Slot-Nr.
1A68: 8A       242 ASL   ;* 16
1A69: 8A       243 ASL
1A6A: 8A       244 ASL
1A6B: 8A       245 ASL
1A6C: A8       246 TAY
1A6D: 68       247 RTS

```

```

249 *
250 *****
251 *
252 * PPOUT - Ausgabe eines Zeichens auf dem Drucker
253 *
254 * Eingabeparameter: PAR1H - Slot-Nr.
255 *                   PAR3L - Ausgabe Zeichen
256 *
257 *****
258 *
1A4E: 20 8F 1A 259 PPOUT JSR PPST1 ;Status Abfrage vorbereiten
260 *
1A71: 81 FA 261 PPSTL1 LDA (PAR1L),Y ;Drucker bereit ?
1A73: 30 FC 262 BMI PPSTL1 ;Nein -)
263 *
1A75: 20 66 1A 264 JSR X16 ;Slot * 16 bestimmen
265 *
266 * Zeichen ausgeben
267 *
1A78: A5 FE 268 LDA PAR3L ;Ausgabe Zeichen
1A7A: 99 80 C8 269 STA IOBASE,Y
1A7D: 60 270 RTS
271 *
272 *****
273 *
274 * PPSTAT - Parallel Printer Ausgabe Status Abfrage
275 *
276 * Eingangsparameter: PAR1H - Slot-Nr.
277 *
278 * Ausgangsparameter: PAR2H - Status (8=nicht -, $FF=bereit)
279 *
280 *****
281 *
1A7E: 20 8F 1A 282 PPSTAT JSR PPST1
283 *
1A81: 81 FA 284 LDA (PAR1L),Y ;Status Port
1A83: 8A 285 ASL
286 *
1A84: A9 FF 287 LDA #$FF ;C=1: nicht bereit
1A86: 69 80 288 ADC #8
289 *
1A8B: 85 FD 290 STRTS STA PAR2H
1A8A: 60 291 PPINI RTS ;keine Initialisierung
292 *
293 *****
294 *
295 * PPIN - scheinbare "Eingabe" vom Drucker
296 *
297 * Eingangsparameter: PAR1H - Slot-Nr.
298 *
299 * Ausgangsparameter: PAR2H - $1A (CTRL-2)
300 *
301 *****
302 *
1A8B: A9 1A 303 PPIN LDA #$1A
1A8D: 08 F9 304 BNE STRTS

```

```

306 *
307 *****
308 *
309 *   PPST1 - Parallel Printer Status Abfrage vorbereiten
310 *
311 *****
312 *
1A8F: A9 C8 313 PPST1   LDA  #C8
1A91: 85 FB 314         ORA  PAR1H
1A93: 85 FB 315         STA  PAR1H
316 *
1A95: A9 88 317         LDA  #8
1A97: 85 FA 318         STA  PAR1L
1A99: A8 C1 319         LDY  #C1
1A9B: 68    320         RTS
321 *
322 *****
323 *

```

--END ASSEMBLY--

ERRORS: 0

193 BYTES

SYMBOL TABLE - ALPHABETICAL ORDER:

CSWH	=#37	IN88Z	=#C840	INI88Z	=#C888	? INIT88	=#1908
? INP88	=#19E7	IOBASE	=#C888	IOSCR4	=#8678	IOSCR5	=#86F8
KSWH	=#39	MAKKU	=#45	MREGX	=#46	MREGY	=#47
OUT88Z	=#C9AA	? OUTP88	=#19F5	PAR1H	=#FB	PAR1L	=#FA
PAR2H	=#FD	PAR3L	=#FE	? PPIN	=#1A8B	? PPINI	=#1A8A
? PPOUT	=#1A6E	PPST1	=#1A8F	? PPSTAT	=#1A7E	PPSTL1	=#1A71
ROMINI	=#1A87	ROMOFF	=#CFFF	SELROM	=#1A18	? SIIN	=#1A47
? SIINI	=#1A28	? SIOUT	=#1A56	? S1STAT	=#1A39	STRTS	=#1A88
WAIT	=#1A4A	WAIT2	=#1A59	X16	=#1A66		

SYMBOL TABLE - NUMERICAL ORDER:

CSWH	=#37	KSWH	=#39	MAKKU	=#45	MREGX	=#46
MREGY	=#47	PAR1L	=#FA	PAR1H	=#FB	PAR2H	=#FD
PAR3L	=#FE	IOSCR4	=#8678	IOSCR5	=#86F8	? INIT88	=#1908
? INP88	=#19E7	? OUTP88	=#19F5	ROMINI	=#1A87	SELROM	=#1A18
? SIINI	=#1A28	? S1STAT	=#1A39	? SIIN	=#1A47	WAIT	=#1A4A
? SIOUT	=#1A56	WAIT2	=#1A59	X16	=#1A66	? PPOUT	=#1A6E
PPSTL1	=#1A71	? PPSTAT	=#1A7E	STRTS	=#1A88	? PPINI	=#1A8A
? PPIN	=#1A8B	PPST1	=#1A8F	IOBASE	=#C888	INI88Z	=#C888
IN88Z	=#C840	OUT88Z	=#C9AA	ROMOFF	=#CFFF		



## Anhang B

Tabellen zur Ein-/Ausgabe Umcodierung im AP 22 BIOS

Mit Hilfe der hier angegebenen Tabellen codiert das AP 22 BIOS eingegebene und ausgegebene Zeichen um. In der hier abgedruckten Tabelle sind die Standard Umcodierungen, wie sie im ausgelieferten System eingebaut sind, angegeben. Bei Ausgabe auf dem Apple 24x40 Zeichen Bildschirm (wird als Slot 0 interpretiert) und bei Ausgabe auf einer Peripheriekarte im Slot 3 werden die Steuerzeichen eines SOROC IQ 120 Terminals unterstützt. Bei Eingabe von der Apple Tastatur oder bei Eingabe von einer Peripheriekarte in Slot 3 werden die auf der Apple Tastatur fehlenden Zeichen und mehrere BASIC-Befehle unterstützt. Dies ist sicher nicht immer eine optimale Lösung. Eigene Umcodierungen koennen durch Aenderung dieser Tabelle eingebaut werden. Die Tabelle finden Sie als Datei UMCODE.ASM auf der AP 22 - CP/M PATCHER Diskette.

BITTE AENDERN SIE NICHT DIE ORIGINAL DATEI - BENUTZEN SIE FUER AENDERUNGEN ZUNAECHST IMMER EINE KOPIE - ES LOHNT SICH !

Die Aenderungen koennen mit einem unter CP/M lauffaehigen Text Editor, wie z. B. ED.COM oder WORDSTAR, vorgenommen werden. Die Beschreibung des CP/M Assemblers ASM ist zu beachten! Nach der Aenderung muessen die folgenden Kommandos eingegeben werden um die neuen Umcodierungen nutzen zu koennen (es wird angenommen das sich ASM.COM, PIP.COM, IBSCPM64.COM und DDT.COM auf Laufwerk A: und UMCODE.ASM auf P: befindet) :

```
A>PIP P:UMCODE.ASM:P:UMCODE.ASM[2]
A>ASM UMCODE.PPX
--- Das Listing wird auf der Console ausgegeben
A>P:
P>A:DDT A:IBSCPM64.COM
DDT VERS 2.2
--- Die Datei IBSCPM64.COM (das CP/M System) wird eingelesen
NEXT PC
3900 0100
-UMCODE.HEX
-R
--- Die Umcodier Tabelle wird eingelesen
NEXT PC
3900 0100
-G
--- Sie verlassen DDT
P>SAVE S: A:IBSCPM64.COM
--- Das alte System wird durch das geaenderte ersetzt
P>A:IBSCPM64
--- Legen Sie im Laufwerk A: eine Diskette ein und quittieren Sie die Anfrage
--- mit J
--- Das geaenderte System wird jetzt auf diese Diskette geschrieben
P>
--- Durch Aus- und Einschalten des Rechners kann das geaenderte System gestartet
--- werden.
```

Ihre Eingaben sind unterstrichen, mit --- beginnende Zeilen enthalten Kommentare.

## UMCODIER TABELLEN FUER IBS CP/M 810S

DIESE TABELLEN KOENNEN MIT ASM.COM UEBERSETZT WERDEN.  
SIE DIENEN ZUR UMCODIERUNG DER EIN/AUSGABE ZEICHEN

NACH DER ASSEMBLIERUNG KOENNEN DIE TABELLEN MIT  
HILFE DES PROGRAMMS DOT.COM IN DAS SYSTEM INTEGRIERT  
WERDEN. DAZU SIND FOLGENDE BEFEHLE EINZUGEBEN :

```

DDT IBSCPM64.COM      DDT.COM STARTEN UND SYSTEM LADEN
UMCODE.HEX           ASSEMBLER AUSGABE VON UMCODE.ASM
R
G0
SAVE 56 IBSCPM64.COM  WARMER SYSTEM START
                       NEUES SYSTEM RETTEN
  
```

DURCH STARTEN VON IBSCPM64.COM KANN DAS SYSTEM JETZT  
AUF DIE DISKETTE GESCHRIEBEN WERDEN.

8C00

ORG

8C00H

;ADRESSE IN IBSCPM64.COM

## ADRESSTABELLE FUER EINGABE UMCODIERUNG

```

8C00 3E0C
8C0A 0000
8C0C 0000
8C0E 3E0C
8C10 0000
8C12 0000
8C14 0000
8C16 0000
  
```

```

DW      UMITAS      ;TASTATUR UMCODIERUNG
DW      0            ;SLOT 1: KEINE UMCODIERUNG
DW      0            ;SLOT 2: KEINE UMCODIERUNG
DW      UMITAS      ;SLOT 3: TASTATUR UMCODIERUNG
DW      0            ;SLOT 4: KEINE UMCODIERUNG
DW      0            ;SLOT 5: KEINE UMCODIERUNG
DW      0            ;SLOT 6: KEINE UMCODIERUNG
DW      0            ;SLOT 7: KEINE UMCODIERUNG
  
```

## ADRESSTABELLE FUER AUSGABE UMCODIERUNG

```

8C18 200C
8C1A 0000
8C1C 0000
8C1E 200C
8C20 0000
8C22 0000
8C24 0000
8C26 0000
  
```

```

DW      LMOTB       ;40 ZEICHEN UMCODIERUNG
DW      0            ;SLOT 1: KEINE UMCODIERUNG
DW      0            ;SLOT 2: KEINE UMCODIERUNG
DW      LMOTB       ;SLOT 3: 00 ZEICHEN UMCODIERUNG
DW      0            ;SLOT 4: KEINE UMCODIERUNG
DW      0            ;SLOT 5: KEINE UMCODIERUNG
DW      0            ;SLOT 6: KEINE UMCODIERUNG
DW      0            ;SLOT 7: KEINE UMCODIERUNG
  
```

## AUSGABE UMCODIER TABELLE

## AUFBAU DER UMCODIER TABELLEN :

(S. SOFTCARD BESCHREIBUNG VOLUME I SEITE 2-12 BIS 2-16)

## DIREKT AUF EINANDERFOLGEND FUER SOFT- UND HARD TAB. :

```

BYTE 0 : POSITIONIER OFFSET
BYTE 1 : EINFUEHRUNGS ZEICHEN (LEAD IN)
BYTE 2 : BILDSCHIRM LOESCHEN #1
BYTE 3 : BIS BILDSCHIRME LÖSCHEN #2
BYTE 4 : BIS ZEILENENDE LÖSCHEN #3
BYTE 5 : TEXTDARSTELLUNG WEISS AUF SCHWARZ #4
BYTE 6 : TEXTDARSTELLUNG SCHWARZ AUF WEISS #5
BYTE 7 : CURSOR OBEN LINKS POSITIONIEREN #6
BYTE 8 : CURSOR ABSOLUT POSITIONIEREN #7
BYTE 9 : CURSOR 1 ZEILE AUFWAERTS #8
BYTE 10 : CURSOR 1 STELLE NACH RECHTS #9

```

```

0C20 20      UMTB:  DB      20H      ;OFFSET SOFTWARE
0C29 1B      DB      019H      ;LEAD-IN = ESC

0C2A AA      DB      '*' +128    ;ESC * = BILD LOESCHEN
0C2B 09      DB      'Y' +128    ;ESC Y = CLEAR TO EOS
0C2C 04      DB      'T' +128    ;ESC T = CLEAR TO EOL
0C2D A9      DB      ')' +128    ;ESC ) = NORMALDARSTELLUNG
0C2E A8      DB      '(' +128    ;ESC ( = INVERSE DARSTELLUNG
0C2F 1E      DB      !EH      ;RS = CURSOR HOME
0C30 8D      DB      '=' +128    ;ESC = = ABSOLUT POSITIONIEREN
0C31 8B      DB      0BH      ;UT = CURSOR UP
0C32 8C      DB      0CH      ;FF = CURSOR RECHTS

```

## HARDWARE

```

0C33 A0      DB      0A0H      ;OFFSET XY 32
0C34 00      DB      0        ;KEIN LEAD-IN

0C35 8C      DB      0CH      ;FF = BILD LOESCHEN
0C36 8B      DB      0BH      ;UT = CLEAR TO EOS
0C37 1D      DB      1DH      ;GS = CLEAR TO EDL
0C38 8E      DB      0EH      ;SO = NORMALDARSTELLUNG
0C39 0F      DB      0FH      ;SI = INVERSE DARSTELLUNG
0C3A 19      DB      19H      ;EM = CURSOR HOME
0C3B 1E      DB      1EH      ;RS = ABSOLUT POSITIONIEREN
0C3C 1F      DB      1FH      ;US = CURSOR UP
0C3D 1C      DB      1CH      ;FS = CURSOR RECHTS

```

## TASTATUR EINGABE UMCODIER-TABELLE

DIE EINGABE UMCODIER TABELLE IST WIE FOLGT AUFGEBAUT :

&lt;..&gt; = KANN WIEDERHOLT WERDEN, [..] = OPTIONAL

```

[ (
  BYTE 0 : EINLEITUNGS (LEAD-IN) ZEICHEN
  BYTE 1 : LSB (ZUGEHÖRIGE EINGABE TABELLE)
  BYTE 2 : MSB (ZUGEHÖRIGE EINGABE TABELLE) ) ]

```

```

  BYTE 0 : 0FFH (ENDEKENNUNG FUER LEAD-IN)
  BYTE 1 : ANZAHL DER UMCODIER ZEICHEN

```

ANZAHL\*(BYTE 0 : UMZUCODIERENDES EINGABEZEICHEN )

ANZAHL\*(BYTE 0 : OFFSET IN WEITERGABEZEICHEN TAB. )

WEITERGABE-TABELLE (SCHLIESST DIREKT AN):

```

ANZAHL*(BYTE 0 : ANZ2 DER WEITERGABE ZEICHEN
ANZ2*(BYTE 0 : WEITERGABE ZEICHEN)

```

BEM.: OFFSET = 0 : 1. ANZ2-ANGABE IN WEITERGABE-TAB.  
 OFFSET ZEIGT AUF DIE ANZ2-ANGABE DER ZUM LEAD-  
 IN ZEICHEN GEHÖRENDE WEITERZUGEBENDE ZEICHEN.  
 DIE WEITERGABE-TAB. IST BESTANDTEIL DER EINGABE-  
 TABELLE.

```

.....
9C3E 1B          UMITAS: DB      1BH          ;LEAD IN : CTRL-[ (ESC) (
9C3F 470C        DW      ATB1          ;ZUGEHÖRIGE TABELLE
;
9C41 FF          ;          DB      0FFH          ;LEAD IN ENDE
;
9C42 01          ;          DB      1          ;1 ZEICHEN UMCODIEREN
9C43 0B          ;          DB      ['+129
;
9C44 00          ;          DB      0          ;OFFSET IN WEITERGABE TABELLE
;
WEITERGABE-TABELLE ZU UMITAS
9C45 01          DB      1          ;LAENGE
9C46 00          DB      0BH          ;CTRL-K

```

```

:
:-----
:
:          UMCODIERTABELLE MIT LEAD-IN ESC
:
9C47 FF      AT91:  DB      0FFH          ;KEINE WEITEREN LEAD-IN
:
9C48 12      :      DB      CHNUM        ;ANZAHL UMCODIER-ZEICHEN
:
:          ERKANNTE ZEICHEN
:
9C49 CC      CHANF:  DB      'L'+128     ;ESC L -> LOAD*
9C4A 03      DB      'S'+128     ;ESC S -> SAVE*
9C4B 02      DB      'R'+128     ;ESC R -> RUN*
9C4C CB      DB      'K'+128     ;ESC K -> KILL*
9C4D C6      DB      'F'+128     ;ESC F -> FILES
9C4E CD      DB      'M'+128     ;ESC M -> MBASIC
9C4F 08      DB      'X'+128     ;ESC X -> SYSTEM (CR)
9C50 41      DB      'A'         ;ESC A -> [ (AE)
9C51 61      DB      'a'         ;ESC a -> ( (ae)
9C52 4F      DB      'O'         ;ESC O -> \ (OE)
9C53 6F      DB      'o'         ;ESC o -> ! (oe)
9C54 0F      DB      0FFH        ;ESC CTRL-0 -> CTRL-] (GS)
9C55 55      DB      'U'         ;ESC U -> ] (UE)
9C56 75      DB      'u'         ;ESC u -> } (ue)
9C57 08      DB      9          ;ESC BS -> RUB
9C58 2D      DB      '-'         ;ESC - -> _ (Underline)
9C59 3F      DB      '?'         ;ESC ? -> CTRL-_ (US)
9C5A D1      DB      'Q'+128     ;ESC Q -> CTRL-A
:
0812 =      :      CHNUM  EQU      1-CHANF   ;ANZAHL DER UMCODIER-ZEICHEN
:
:          OFFSETS IN WEITERGABE-TABELLE :
:
9C5B 08      DB      9
9C5C 86      DB      SAVE-WSTRT
9C5D 0C      DB      RUN-WSTRT
9C5E 11      DB      KILL-WSTRT
9C5F 17      DB      FILES-WSTRT
9C60 1D      DB      MBAS-WSTRT
9C61 24      DB      SYS-WSTRT
9C62 2C      DB      GAE-WSTRT
9C63 2E      DB      KAE-WSTRT
9C64 38      DB      GOE-WSTRT
9C65 32      DB      KOE-WSTRT
9C66 34      DB      GS-WSTRT
9C67 36      DB      GUE-WSTRT
9C68 38      DB      KUE-WSTRT
9C69 3A      DB      RUB-WSTRT
9C6A 3C      DB      ULI-WSTRT
9C6B 3E      DB      US-WSTRT
9C6C 48      DB      CTLA-WSTRT

```



Anhang CTabelle zur Ein- / Ausgabe Zuordnung

Mit der hier angegebenen Tabelle wird die Verbindung zwischen den Text Ein- / Ausgabe Peripheriekarten in den Apple II Slots (bzw. den eingebauten E/A) und den CP/M Geräeten hergestellt. Waehrend die Zuordnung der logischen zu den physikalischen Geräeten im CP/M Betriebssystem, ueber das IOBYTE, festgelegt ist und somit auch nicht geaendert werden sollte, kann die Abbildung auf die Slots und interne E/A frei, den jeweiligen Anforderungen entsprechend, gewaehlt werden. Dazu muessen die "Treiber Additionswerte fuer Slots" in der Zuordnungstabelle, evtl. auch die dort angegebenen 288 Treiber Adressen, geaendert werden. Wird z. B. "TTYIN EQU UNIINP+SLOT3" in "TTYIN EQU UNIINP+SLOT2" geaendert, so erfolgt die Eingabe ueber das physikalische Geräet TTY von einer Peripheriekarte in Slot 2 anstelle von Slot 3. Ist in einem Slot keine Peripheriekarte installiert, so erfolgen Eingaben ueber die Apple Tastatur und Ausgaben auf dem 24x40 Zeichen Bildschirm, bei Status Abfragen wird dann immer "nicht bereit" uebergeben. Die Zuordnungstabelle finden Sie als Datei E/AZUORD.ASM auf der AP 22 - CP/M PATCHER Diskette.

FUEHREN SIE AENDERUNGEN ZUNAECHST IMMER AUF EINER KOPIE DER MASTER DISKETTE DURCH - BEI \*FALSCHEN\* AENDERUNGEN ERSPAREN SIE SICH VIEL ARBEIT !

Die Aenderungen koennen mit einem unter CP/M lauffaehigen Text Editor, wie z. B. ED oder WORDSTAR, vorgenommen werden, allerdings sollten sie vorher schon einmal mit dem CP/M Assembler ASM gearbeitet, oder aber zumindest dessen Beschreibung gelesen haben. Wenn die neue Tabelle syntaktisch fehlerfrei ist, kann sie zunaechst einmal direkt getestet werden, dazu muessen die folgenden Kommandos eingegeben werden (es wird angenommen das sich die Dateien ASM.COM, DOT.COM, PIP.COM und IBSCPM64.COM auf Laufwerk A: und E/AZUORD.ASM auf Laufwerk P: befindet):

A) PIP P:E/AZUORD.ASM=P:E/AZUORD.ASM(7)  
 --- Bit 7 des Textes wird gelöscht um Fehler zu vermeiden  
 --- **WICHTIG:** Die geänderte Tabelle muss sich auch auf einer Diskette befinden,  
 --- da sie sonst bei einem Fehler verlorengehen kann !!

A) ASM E/AZUORD.PP2  
 CP/M ASSEMBLER VER 2.0  
 --- Die geänderte Tabelle wird assembliert, als Ergebnis wird die neue Tabelle  
 --- in der Datei P:E/AZUORD.HEX abgelegt.  
 F7AB  
 001H USE FACTOR  
 END OF ASSEMBLY

--- Es gibt jetzt 2 Möglichkeiten: 1. testen der geänderten Tabelle im  
 --- laufenden System oder 2. System modifizieren und auf die Boot-Spuren  
 --- schreiben und dann nach einem Neustart testen.

--- Möglichkeit 1: Direktes Testen

A) DDT P:E/AZUORD.HEX  
 --- Das laufende System wird geändert  
 A)  
 --- Prüfen Sie jetzt ob ihre Änderung funktioniert ...  
 --- wenn Ja, machen Sie mit Möglichkeit 2 weiter

--- Möglichkeit 2: System ändern und dann testen

A) P:  
 P) A:DDT A:IBSCPM64.COM  
 DDT VERS 2.2  
 --- Die Datei IBSCPM64.COM (das CP/M System) wird eingelesen

NEXT PC  
 3900 0100  
 -I/AZUORD.HEX  
 -R3C00  
 --- Die geänderte Zuordnungstabelle wird eingelesen

NEXT PC  
 3900 0100

-00  
 --- Sie verlassen DDT

P) SAVE 56 A:IBSCPM64.COM  
 --- Das alte System wird durch das geänderte System ersetzt

A) P:  
 A) PIP A:P:E/AZUORD.ASM(U)  
 --- Die geänderte Tabelle wird gesichert

DESTINATION IS R/O, DELETE (Y/N)?Y  
 --- Diese Anfrage erfolgt nur wenn E/AZUORD.ASM auf Laufwerk A:  
 --- schreibgeschützt ist

A)  
 --- Durch Aus- und Einschalten des Rechners kann das geänderte System gestartet  
 --- werden.

Ihre Eingaben sind unterstrichen, mit --- beginnende Zeilen enthalten Kommentare.

```

*****
*   Ein- / Ausgabe Zuordnung im AP 22 8105   *
*

```

```

*   Version 1.01 - 26. 7. 1983   *
*

```

```

*   Copyright (C) 1983 by Rainer Ellerbrake *
*
*****

```

```

-----
KONSTANTEN
-----

```

```

Treiber Additionswerte fuer Slots

```

```

;
;   Um eine Peripheriekarte in einem Slot anzusprechen
;   muss die Slot-Nr. * 256 zur 6502 Treiber-Nr. hinzu-
;   addiert werden. Slot 0 steht fuer die Apple Tastatur
;   bzw. den Apple 24x48 Zeichen Bildschirm.
;

```

```

;
;   0100 = SLOT1 EQU 100H
;   0200 = SLOT2 EQU 200H
;   0300 = SLOT3 EQU 300H
;   0400 = SLOT4 EQU 400H
;   0500 = SLOTS EQU 500H
;   0600 = SLOT6 EQU 600H
;   0700 = SLOT7 EQU 700H
;

```

```

-----
;
;   6502 Treiber Nummern
;

```

```

;   Die 6502 Treiber Nummern stehen stellvertretend fuer die
;   Adresse eines Treibers im Apple Speicher. Die hier vor-
;   gegebenen Nummern stehen fuer universelle Treiber die
;   ihrerseits erst in abhaengigkeit von der Slot-Nr., den
;   tatsaechlichen Peripherietreiber aufrufen.
;

```

```

;   DIESE NUMMERN BLEIBEN UNVERAENDERT WENN NEUE TREIBER INS
;   BIOS EINGEBUNDEN WERDEN !
;

```

```

;
;   0811 = UNIINP EQU 17 ;Universal Eingabe Treiber
;   0812 = UNIOUT EQU 18 ;Universal Ausgabe Treiber
;   0813 = UNISTAT EQU 19 ;Universal Status Treiber
;

```

.....

288 Treiber Adressen

Anstelle eines 6502 Treibers kann auch ein 288 Treiber angegeben werden.

F702 =	CRTSTAT	EQU	8F7D2H		
F706 =	STADUM	EQU	8F7D4H		;Apple Tastatur Status
F708 =	INDUM	EQU	8F7D8H		;Status nicht vorhanden
F70A =	OUTDUM	EQU	8F7DAH		;Eingabe nicht vorhanden (liest ^Z)
					;Ausgaben werden ignoriert

.....

Andere Konstanten

8888 =	MULPROC	EQU	128		;6502 Treiber (aeuft parallel)
--------	---------	-----	-----	--	--------------------------------

.....

Zuordnung der physikalischen CP/M Gerate zu Slots und eingebauter Ein- und Ausgabe

Durch die nachfolgenden Definitionen werden die Einzel-  
funktionen der physikalischen CP/M Gerate festgelegt.  
Die Funktion wird durch die Treiber Adresse bzw. der  
6502 Treiber-Nr. bestimmt; im 2. Fall kann zusaetzlich  
der Slot, auf dem sie ausgefuehrt werden soll, angegeben  
werden.

.....

Console-faehige Gerate

Die hier definierten physikalischen Gerate koennen den  
CONSOLE zugeordnet werden.

8311 =	TTYIN	EQU	UNIINP+SLOT3		
8392 =	TTYOUT	EQU	UNIOUT+MULPROC+SLOT3		;TTY: Eingabe
F702 =	TTYSTAT	EQU	CRTSTAT		;TTY: Ausgabe
					;TTY: Status (Apple Tastatur)
8311 =	CRTIN	EQU	UNIINP+SLOT3		
8392 =	CRTOUT	EQU	UNIOUT+MULPROC+SLOT3		;CRT: Eingabe
			CRTSTAT s. 288 Treiber Adressen		;CRT: Ausgabe
8311 =	UCIIN	EQU	UNIINP+SLOT3		
8192 =	UCIOUT	EQU	UNIOUT+MULPROC+SLOT1		;UCI: Eingabe
F702 =	UCI:STAT	EQU	CRTSTAT		;UCI: Ausgabe
					;UCI: Status (Apple Tastatur)

.....

Reader Eingabe (RDR:)

Die hier definierten Gerate koennen dem logischen CP/M Gerat RDR: zugeordnet werden.

0211 =	PTRIN	EQU	UNIINP+SLOT2	;PTR: Eingabe
F708 =	UR1IN	EQU	INDUM	;UR1: Eingabe
F708 =	UR2IN	EQU	INDUM	;UR2: Eingabe

.....

Puncher Ausgabe (PUN:)

Die hier definierten Gerate koennen dem logischen CP/M Gerat PUN: zugeordnet werden.

0292 =	PTP0UT	EQU	UNI0UT+MULPROC+SLOT2	;PTP: Ausgabe
F70A =	UP10UT	EQU	OUTDUM	;UP1: Ausgabe
F70A =	UP20UT	EQU	OUTDUM	;UP2: Ausgabe

.....

List Ausgabe (LST:)

Die hier definierten Gerate koennen dem logischen CP/M Gerat LST: zugeordnet werden.

0192 =	LPT0UT	EQU	UNI0UT+MULPROC+SLOT1	;LPT: Ausgabe
0113 =	LPTSTAT	EQU	UNISTAT+SLOT1	;LPT: Status
F70A =	UL10UT	EQU	OUTDUM	;UL1: Ausgabe
F706 =	UL1STAT	EQU	STADUM	;UL1: Status

.....

Adress Tabelle fuer die Gerate Treiber

In dieser Tabelle werden :

1. die Einzelfunktionen der physikalischen CP/M Gerate definiert und
2. jeder Funktion eines logischen CP/M Gerats die Funktionen der physikalischen CP/M Gerate zugeordnet, auf die das logische CP/M Gerat durch das IOBYTE gelegt werden kann.

DIE FOLGENDE TABELLE SOLLTE NICHT VERAEENDERT WERDEN !

Die tatsaechliche Anfangsadresse der Tabelle steht im AP 22 Speicher in Adresse 0FFCEH und 0FFCFH (LSB, MSB). Die dort angegebene Adresse ist massgebend !!

.....

F77A

ORG	0F77AH	;NUR fuer 810S Vers. 1.0!
-----	--------	---------------------------

Logisches Gerat : CON:  
 Phys. Gerate : TTY:, CRT:, BAT:, UC1:

## CON: Eingabe Status Treiber

F77A D2F7	DW	TTYSTAT	:TTY:
F77C D2F7	DW	CRTSTAT	:CRT:
F77E D6F7	DW	STADUM	:BAT: liefert keinen Status
F788 D2F7	DW	UC1STAT	:UC1:

## CON: Eingabe Treiber

F782 1183	DW	TTYIN	:TTY:
F784 1183	DW	CRTIN	:CRT:
F786	DS	2	:BAT: ist kein phys. Gerat
F788 1183	DW	UC1IN	:UC1:

## CON: Ausgabe Treiber

F78A 9283	DW	TTYOUT	:TTY:
F78C 9283	DW	CRTOUT	:CRT:
F78E	DS	2	:BAT: ist kein phys. Gerat
F798 9281	DW	UC1OUT	:UC1:

Logisches Gerat : RDR:  
 Phys. Gerate : TTY:, PTR:, UR1:, UR2:

## RDR: Eingabe Treiber

F792 1183	DW	TTYIN	:TTY:
F794 1182	DW	PTRIN	:PTR:
F796 D8F7	DW	UR1IN	:UR1:
F798 D8F7	DW	UR2IN	:UR2:

```

:-----:
:
:   Logisches Gerat : PLN:
:   Phys. Gerate   : TTY:, PTP:, UP1:, UP2:
:-----:
:
:   PLN: Ausgabe Treiber
:
:   F79A 9283      OW      TTYOUT      ;TTY:
:   F79C 9282      OW      PTPOUT      ;PTP:
:   F79E DAF7      OW      UP1OUT      ;UP1:
:   F7A8 DAF7      OW      UP2OUT      ;UP2:
:-----:
:
:   Logisches Gerat : LST:
:   Phys. Gerate   : TTY:, CRT:, LPT:, UL1:
:-----:
:
:   LST: Ausgabe Status Treiber
:
:   F7A2 D6F7      OW      STADUM      ;TTY: hat keinen Ausgabe Status
:   F7A4 D6F7      OW      STADUM      ;CRT: hat keinen Ausgabe Status
:   F7A6 1381      OW      LPTSTAT     ;LPT:
:   F7A8 D6F7      OW      UL1STAT     ;UL1:
:-----:
:
:   LST: Ausgabe Treiber
:
:   F7AA 9283      OW      TTYOUT      ;TTY:
:   F7AC 9283      OW      CRTOUT      ;CRT:
:   F7AE 9281      OW      LPTOUT      ;LPT:
:   F7B8 DAF7      OW      UL1OUT      ;UL1:
:-----:
:
:   F7B2          END

```

1. Name of the person or organization  
2. Address

3. Details of the work done

1973	1974	1975
1000	1000	1000
1000	1000	1000
1000	1000	1000

1000  
1000  
1000  
1000

4. Name of the person or organization  
5. Address

6. Details of the work done

1973	1974	1975
1000	1000	1000
1000	1000	1000
1000	1000	1000

1000  
1000  
1000  
1000

(DIESE SEITE BLEIBT LEER)

1973	1974	1975
1000	1000	1000
1000	1000	1000
1000	1000	1000

1000  
1000  
1000  
1000

Anhang DTabelle zur Disketten Zuordnung

Mit der hier angegebenen Tabelle wird die Verbindung zwischen den Disk Controller Peripheriekarten in den Apple II Slots und deren Laufwerken und den CP/M Laufwerken hergestellt. Jedem vorhandenen Laufwerk kann eines der 4 CP/M Laufwerke zugeordnet werden. In dieser Tabelle kann nur 1 Byte frei definiert werden, die anderen 4 Byte eines Eintrags werden entweder aus anderen Tabellen eingesetzt oder aber sind vorgegeben.

**FUEHREN SIE AENDERUNGEN ZUNAECHST IMMER AUF EINER KOPIE DER MASTER DISKETTE DURCH - BEI \*FALSCHEN\* AENDERUNGEN ERSPAREN SIE SICH VIEL ARBEIT !**

Die Aenderungen koennen mit einem unter CP/M lauffaehigen Text Editor, wie z. B. ED oder WORDSTAR, vorgenommen werden, allerdings sollten sie vorher schon einmal mit dem CP/M Assembler ASM gearbeitet, oder aber zumindest dessen Beschreibung gelesen haben. Wenn die neue Tabelle syntaktisch fehlerfrei ist, kann sie zunaechst einmal direkt getestet werden, dazu muessen die folgenden Kommandos eingegeben werden (es wird angenommen das sich die Dateien ASM.COM, DDT.COM, PIP.COM und IBSCPM64.COM auf Laufwerk A: und DSKZUORD.ASM auf Laufwerk P: befindet):

A) PIP P:DSKZUORD.ASM=P:DSKZUORD.ASM(Z)

--- Bit 7 des Textes wird gelöscht um Fehler zu vermeiden

--- **WICHTIG:** Die geänderte Tabelle muss sich auch auf einer Diskette befinden,  
--- da sie sonst bei einem Fehler verlorengehen kann !!

A) ASM DSKZUORD.PPZ

CP/M ASSEMBLER VER 2.0

--- Die geänderte Tabelle wird assembliert, als Ergebnis wird die neue Tabelle  
--- in der Datei P:DSKZUORD.HEX abgelegt.

0831

081H USE FACTOR

END OF ASSEMBLY

--- Die geänderte Tabelle wird jetzt in das System integriert

A) P:

P) A:DOT A:IBSCPM64.COM

DOT VERS 2.2

--- Die Datei IBSCPM64.COM (das CP/M System) wird eingelesen

NEXT PC

3900 0100

-IDSKZUORD.HEX

-R

--- Die geänderte Zuordnungstabelle wird eingelesen

NEXT PC

3900 0100

-G8

--- Sie verlassen DOT

P) SAVE S6 A:IBSCPM64.COM

--- Das alte System wird durch das geänderte System ersetzt

P) A:

A) PIP A:=P:DSKZUORD.ASM(U)

--- Die geänderte Tabelle wird gesichert

DESTINATION IS R/O, DELETE (Y/N)?Y

--- Diese Anfrage erfolgt nur wenn DSKZUORD.ASM auf Laufwerk A:  
--- schreibgeschützt ist

A)

--- Durch Aus- und Einschalten des Rechners kann das geänderte System gestartet  
--- werden.

Ihre Eingaben sind unterstrichen, mit --- beginnende Zeilen enthalten Kommentare.





Anhang ETabelle zur Peripheriekarten Erkennung

Die hier angegebene Tabelle wird vom automatischen Kartenerkennungssystem des AP 22 BIOS benutzt um Peripheriekarten mit einem ROM im "I/O ROM Space" zu erkennen und ins System einzubinden. Dazu koennen die Inhalte von maximal 3 beliebigen Bytes des ROM's angegeben werden anhand derer das BIOS die Karte eindeutig erkennen kann. Zu diesen Bytes gehoert ausserdem eine Kartenkennung, die das BIOS in der Slottabelle eintraegt und eine 'Basis Treibernummer' durch die der Treiber festgelegt wird. Die genaue Bedeutung dieser Werte finden Sie im Abschnitt 4.3. In dieser Tabelle koennen maximal 6 Karten definiert werden. Sie finden die hier abgedruckte Datei mit den standardmaessig eingebauten Werten als Datei KENTAB.ASM auf der AP 22 - CP/M PATCHER Diskette.

**BENUTZEN SIE FUER AENDERUNGEN STETS EINE KOPIE DES ORIGINALS - BEI FEHLERN ESPAREN SIE SICH VIEL ARBEIT !**

Die Aenderungen koennen mit einem unter CP/M lauffaehigen Text Editor, wie z. B. ED oder WORDSTAR, vorgenommen werden, allerdings sollten sie vorher schon einmal mit dem CP/M Assembler ASM gearbeitet, oder aber zumindest dessen Beschreibung gelesen haben. Wenn die neue Tabelle syntaktisch fehlerfrei ist, kann sie zunaechst einmal direkt getestet werden, dazu muessen die folgenden Kommandos eingegeben werden (es wird angenommen das sich die Dateien ASM.COM, DDT.COM, PIP.COM und IBSCPM64.COM auf Laufwerk A: und KENTAB.ASM auf Laufwerk P: befindet):

A)PIP P:KENNTAB.ASM=P:KENNTAB.ASM[?] ]

--- Bit 7 des Textes wird gelöscht um Fehler zu vermeiden  
 --- WICHTIG: Die geänderte Tabelle muss sich auch auf einer Diskette befinden,  
 --- da sie sonst bei einem Fehler verlorengehen kann !!

A)ASH KENNTAB.PPZ

CP/M ASSEMBLER VER 2.0

--- Die geänderte Tabelle wird assembliert, als Ergebnis wird die neue Tabelle  
 --- in der Datei P:KENNTAB.HEX abgelegt.

BAES

001H USE FACTOR

END OF ASSEMBLY

--- Das System wird jetzt geändert

A)P1

P>A:DOT A:IBSCPM64.COM

DOT VERS 2.2

--- Die Datei IBSCPM64.COM (das CP/M System) wird eingelesen

NEXT PC

3900 0100

-KENNTAB.HEX

-R

--- Die geänderte Zuordnungstabelle wird eingelesen

NEXT PC

3900 0100

-GB

--- Sie verlassen DOT

P>SAVE S6 A:IBSCPM64.COM

--- Das alte System wird durch das geänderte System ersetzt

P>A1

A)PIP A:=P:KENNTAB.ASM[U]

--- Die geänderte Tabelle wird gesichert

DESTINATION IS R/O, DELETE (Y/N)?Y

--- Diese Anfrage erfolgt nur wenn KENNTAB.ASM auf Laufwerk A: schreibgeschützt  
 --- ist

A)

--- Durch Aus- und Einschalten des Rechners kann das geänderte System gestartet  
 --- werden.

Ihre Eingaben sind unterstrichen, mit --- beginnende Zeilen enthalten Kommentare.

```

*****
;*
;* Peripheriekarten Erkennungs Tabelle
;*
;* Version 1.8 - 18. 6. 1983
;*
;* Copyright (C) 1983 by Rainer Ellerbrake
;*
*****

```

---

 KONSTANTEN .

---

 6502 Basis Treibernummern

Die 6502 Treiber Nummern stehen stellvertretend fuer die Adresse eines Treibers im Apple Speicher. Die hier angegebenen Nummern sind die Basis Treibernummern fuer die jeweilige Peripheriekarte. Zu einer Text Ein-/Ausgabe Karte gehoeren 4, zu einer Disk Controller Karte 6 Treibernummern.

```

0040 = BASADSK EQU 40H ;Apple II Disk Treiber
0024 = BAS88 EQU 24H ;88 Zeichen Karten Treiber
0028 = BASSI EQU 28H ;Serial Communication Treiber
002C = BASHP EQU 2CH ;Parallel Printer Treiber

```

---

Definition der vom AP 22 BIOS automatisch erkannten Peripheriekarten mit ROM. In dieser Tabelle werden die Erkennungsbytes im ROM, die Kennung der Karte in der Slottabelle und die Basis Treibernummer festgelegt. Die Tabelle darf maximal 6 Eintraege lang sein!

---

```

0AC7 ORG 0AC7H ;Tabellenanfang (Apple Speicher)
0AC7 2B 0B KONFEND-KONFTB ;Tabellensaenge
0AC3 = KONFTB EQU 3

```

---

 Apple II Disk Controller

 Erkennungsbytes: Adr. \$85 - Kennung \$83  
 Adr. \$87 - Kennung \$3C

Kartenkennung: \$88

Basis Treibernr.: \$48

8AC8 48	DB	BASADSK	;Basis Treibernummer
8AC9 88	DB	88H	;Kartenkennung
8ACA 8888	DB	8,8	;3. Kennung: beliebig
8ACC 3C87	DB	3CH, 7	;ROM-Byte 7: \$3C
8ACE 8385	DB	3, 5	;ROM-Byte 5: \$83

---

 88 Zeichen Karte

 Erkennungsbytes: Adr. \$85 - Kennung \$38  
 Adr. \$87 - Kennung \$18

Kartenkennung: \$84

Basis Treibernr.: \$24

8AD8 24	DB	BAS88	;Basis Treibernummer
8AD1 84	DB	4	;Kartenkennung
8AD2 8888	DB	8,8	;3. Kennung: beliebig
8AD4 1887	DB	18H, 7	;ROM-Byte 7: \$18
8AD6 3885	DB	38H, 5	;ROM-Byte 5: \$38

---

 Parallel Printer Interface

 Erkennungsbytes: Adr. \$85 - Kennung \$48  
 Adr. \$87 - Kennung \$48

Kartenkennung: \$85

Basis Treibernr.: \$2C

8AD8 2C	DB	BASHP	;Basis Treibernummer
8AD9 85	DB	5	;Kartenkennung
8ADA 8888	DB	8,8	;3. Kennung: beliebig
8ADC 4887	DB	48H, 7	;ROM-Byte 7: \$48
8ADE 4885	DB	48H, 5	;ROM-Byte 5: \$48





Anhang F6502 Mnemonics Makro Paket fuer Microsoft MACRO-88 Vers. 3.4 u. hoeher

Mit Hilfe dieses Makro Pakets und des Makroassemblers MACRO-88 (M88.COM) der Firma Microsoft koennen 6502 Programme unter CP/M erstellt werden. Dazu muss das Makro-Paket in den Quelltext des Programms einbezogen werden. Dies kann mit Hilfe des Befehls INCLUDE geschehen, als Beispiel ist der Apple 40 Zeichen Ein-/Ausgabe Treiber des AP 22 BIOS hinter dem Makro Paket abgedruckt. Syntaktisch muessen die Programme etwas anders aufgebaut sein:

1. Alle Befehle muessen mit einem . beginnen,
2. bei Immediate Befehlen muss #, statt nur # angegeben werden,
3. fuer Zahlen gilt die Notation des MACRO-88 (z. B. \$100 ist falsch),
4. vor Benutzung eines 6502 Befehls muss .6502 angegeben werden,
5. als Abschluss des 6502 Teils muss .NO6502 angegeben werden,
6. mit Hilfe der Anweisungen .ALLABS und .NORMAL kann absolute Adressierung erzwingen werden (notwendig bei Verwendung von EXTERNAL's und bei Benutzung von Segmenten).

Beispiele:

Normale 6502 Syntax	:	Makro-Paket Syntax
: LDA #55	:	.LDA #,55H
: STA \$2000,X	:	.STA 2000H,X
: EOR (5FC),Y	:	.EOR (8FCH),Y

Alle Assembleranweisungen des MACRO-88 stehen weiterhin zur Verfuegung. 288 und 6502 Programnteile koennen gemischt werden.

Bei Einteilung des Programms in Segmente (CSEG, DSEG, COMMON-Bereiche) und bei der Verwendung von EXTERNAL's ist folgendes zu beachten:

Als Adresse eines innerhalb eines Segments (ausser ASEG) definierten Labels wird, in Abfragen und Verknuepfungen, die Distanz zum Segmentanfang verwendet. Dadurch bedingt wird dessen Adresse vom Makro-Paket faelschlicherweise als Zero-Page Adresse aufgefasst (R und E Fehlermeldungen des MACRO-88 treten auf). Um dies zu verhindern, muss vor dem Befehl die Anweisung .ALLABS stehen, die bewirkt, dass grundsaeztlich Adressen ausserhalb der Zero-Page angenommen werden. Anschliessend kann mit der Anweisung .NORMAL wieder in den Normalmodus zurueckgeschaltet werden.

Die Anzahl der aufgetretenen 6502-Fehler wird bei der Anweisung .NO6502 auf der Console ausgegeben; sie erscheint nicht im Listing!





```

+          ENDIF
+          ENDIF
+          ENDM
+
+          ;-----
+          ;
+          ; ?BRA : GENERIERE RELATIVE SPRUENGE
+          ;
+          ;
+          ; ?BRA MACRO P1, P2
+          ; .CREF
+          ; DEFB P1, P2
+          ; .XCREF
+          ; IF2
+          ; IFF P2 GE 65408 OR (P2 LT 128 AND P2 GE 0)
+          ; .PRINTX /-----> DISTANZ IST ZU GROSS/
+          ; ?ER6502 DEFL ?ER6502 + 1
+          ; ENDIF
+          ; ENDIF
+          ; ENDM
+
+          ;-----
+          ;
+          ; OPER1 : BESTIMME OPERAND TYP !
+          ;
+          ; --> #,IMMEDIATE
+          ; --> ZERO PAGE
+          ; --> ZERO PAGE,X
+          ; --> ABSOLUTE
+          ; --> ABSOLUTE,X
+          ; --> ABSOLUTE,Y
+          ; --> (INDIRECT,X)
+          ; --> (INDIRECT),Y
+          ;
+          ;
+          ; OPER1 MACRO P1,P2
+          ;
+          ; ;
+          ; IFNB (P2)
+          ;
+          ; ;
+          ; OPERANDEN MIT 2 PARAMETERN
+          ;
+          ; ;
+          ; IFION (&#),<P1> ;:#,IMMEDIATE ?
+          ; DEFB ?CODE+8,P2
+          ; EXITH
+          ; ENDIF
+          ;
+          ; ;
+          ; IFION (<X>,<P2> ;:(INDIRECT,X)
+          ; .CREF
+          ; DEFB ?CODE,P1&
+          ; .XCREF
+          ; EXITH
+          ; ENDIF
+          ;
+          ; ;
+          ; IFION (<X>,<P2> ;: ZERO PAGE,X ODER AB
+          ;
+          ; ZERODABS ?CODE+14H, P1
+          ; EXITH
+          ; ENDIF
+          ;
+          ; ;
+          ; IFION (<Y>,<P2> ;: ABSOLUTE,Y ?

```

```

+ ?RES DEFL 0 ;; ODER (INDIRECT),Y
+ 1RFC X,(P1)
+ IF 'BX' EQ ''
+ ?RES DEFL NOT 0
+ ENDF
+ EXITM
+ ENDM
+ ;;
+ KRITERIUM : 1. <
+ IF ?RES
+ .CREF
+ DEFB ?CODE+10H, P1 ;;(INDIRECT),Y
+ .XCREF
+ ELSE
+ DEFB ?CODE+19H ;;ABSOLUT,Y
+ .CREF
+ DEFW P1
+ .XCREF
+ ENDF
+ EXITM
+ ENDF
+ ;;
+ 1F2
+ .PRINTX /-----) SYNTAX FEHLER IN 6582 OPERAND/
+ ?ER6582 DEFL ?ER6582 + 1
+ ENDF
+ ;;
+ ELSE ;; ES FOLGEN WERTE OHNE P2
+ ;;
+ ZEROABS ?CODE+4, P1 ;;ZERO PAGE OR ABSOLUTE
+ ENDF
+ ;;
+ ENDM
+ ;
+ ;-----
+ ;
+ ; OPER2 : BESTIMME OPERAND TYP 2
+ ; --) #,IMMEDIATE (OPER21)
+ ; --) A(accumulator) (OPER2A)
+ ; --) ZERO PAGE
+ ; --) ZERO PAGE,(P3)
+ ; --) ABSOLUTE
+ ; --) ABSOLUTE,(P3)
+ ;
+ * OPER2A MACRO P1,P2
+ * ;;
+ * 1FB (P2)
+ * ;;
+ * OPERAND MIT 1 PARAMETER
+ * ;;
+ * IFDN (A),(P1)
+ * DEFB ?CODE+4
+ * EXITM
+ * ENDF
+ * ENDF
+ * ;;
+ * OPER2 P1,P2,X
+ * ENDM

```







```

;-----;
;
; ASL: 6502 ASL-BEFEHL
;
+ .ASL MACRO P1,P2
+ ;;
+ IF M6502 ;;ONLY IF 6502
+ ?CODE DEFL 006H
+ OPER2A P1,P2
+ ;;
+ ELSE
+ ;;
+ .PRINTX /-----> KEINE 6502-BEFEHLE !!/
+ ?ER6502 DEFL ?ER6502 + 1
+ ENDF
+ ;;
+ ENDM

```

```

;-----;
;
; BCC: 6502 BCC-BEFEHL
;
+ .BCC MACRO P1
+ ;;
+ IF M6502 ;;ONLY IF 6502
+ ?BRA 90H, P1-#-1
+ ;;
+ ELSE
+ ;;
+ .PRINTX /-----> KEINE 6502-BEFEHLE !!/
+ ?ER6502 DEFL ?ER6502 + 1
+ ENDF
+ ;;
+ ENDM

```

```

;-----;
;
; BCS: 6502 BCS-BEFEHL
;
+ .BCS MACRO P1
+ ;;
+ IF M6502 ;;ONLY IF 6502
+ ?BRA 860H, P1-#-1
+ ;;
+ ELSE
+ ;;
+ .PRINTX /-----> KEINE 6502-BEFEHLE !!/
+ ?ER6502 DEFL ?ER6502 + 1
+ ENDF
+ ;;
+ ENDM

```

```

;
;-----
;
;      BEQ:   6502 BEQ-BEFEHL
;
; .BEQ   MACRO P1
;      ;;
;      IF    M6502 ;;ONLY IF 6502
;      ?BRA  8F8H, P1-S-1
;
;      ELSE
;
;      ;;
;      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
;      ?ER6502 DEFL ?ER6502 + 1
;      ENDIF
;
;      ENDM
;
;-----
;
;      BIT:   6502 BIT-BEFEHL
;
; .BIT   MACRO P1
;      ;;
;      IF    M6502 ;;ONLY IF 6502
;      ZERDABS 24H, P1
;
;      ELSE
;
;      ;;
;      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
;      ?ER6502 DEFL ?ER6502 + 1
;      ENDIF
;
;      ENDM
;
;-----
;
;      BMI:   6502 BMI-BEFEHL
;
; .BMI   MACRO P1
;      ;;
;      IF    M6502 ;;ONLY IF 6502
;      ?BRA  38H, P1-S-1
;
;      ELSE
;
;      ;;
;      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
;      ?ER6502 DEFL ?ER6502 + 1
;      ENDIF
;
;      ENDM

```









```

;
;-----
;
;      DEC:   6502 DEC-BEFEHL
;
;      .DEC  MACRO  P1,P2
;      ;;
;      IF    M6502  ;;ONLY IF 6502
;      ?CODE DEFL  8C6H
;      OPER2  P1,P2,X
;
;      ;;
;      ELSE
;
;      ;;
;      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
;      ?ER6502 DEFL  ?ER6502 + 1
;      ENDF
;
;      ;;
;      ENDM
;
;-----

```

```

;
;-----
;
;      DEX:   6502 DEX-BEFEHL
;
;      .DEX  MACRO
;      ;;
;      IF    M6502  ;;ONLY IF 6502
;      DEFB  9CAH
;
;      ;;
;      ELSE
;
;      ;;
;      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
;      ?ER6502 DEFL  ?ER6502 + 1
;      ENDF
;
;      ;;
;      ENDM
;
;-----

```

```

;
;-----
;
;      DEY:   6502 DEY-BEFEHL
;
;      .DEY  MACRO
;      ;;
;      IF    M6502  ;;ONLY IF 6502
;      DEFB  988H
;
;      ;;
;      ELSE
;
;      ;;
;      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
;      ?ER6502 DEFL  ?ER6502 + 1
;      ENDF
;
;      ;;
;      ENDM
;
;-----

```

```

|-----|
|
|      EOR:   6502 EOR-BEFEHL
|
|      .EOR  MACRO  P1,P2
|      ;;
|      IF    M6502  ;;ONLY IF 6502
|      ?CODE DEFL  041H
|      OPER1 P1,<P2>
|
|      ELSE
|
|      ;;
|      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
|      ?ER6502 DEFL  ?ER6502 + 1
|      ENDIF
|      ;;
|      ENDM
|

```

```

|-----|
|
|      INC:   6502 INC-BEFEHL
|
|      .INC  MACRO  P1,P2
|      ;;
|      IF    M6502  ;;ONLY IF 6502
|      ?CODE DEFL  0E6H
|      OPER2 P1,P2,X
|
|      ELSE
|
|      ;;
|      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
|      ?ER6502 DEFL  ?ER6502 + 1
|      ENDIF
|      ;;
|      ENDM
|

```

```

|-----|
|
|      INX:   6502 INX-BEFEHL
|
|      .INX  MACRO
|      ;;
|      IF    M6502  ;;ONLY IF 6502
|      DEFB  0E9H
|
|      ELSE
|
|      ;;
|      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
|      ?ER6502 DEFL  ?ER6502 + 1
|      ENDIF
|      ;;
|      ENDM
|

```

```

;
;-----
;
;      INY:   6502 INY-BEFEHL
;
;      .INY  MACRO
;      ;;
;      IF    M6502   ;;ONLY IF 6502
;      DEFB  80BH
;      ;;
;      ELSE
;      ;;
;      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
;      ?ER6502 DEFL  ?ER6502 + 1
;      ENDIF
;      ;;
;      ENDM
;
;-----
;
;      JMP:   6502 JMP-BEFEHL
;
;      .JMP  MACRO  P1
;      ;;
;      IF    M6502   ;;ONLY IF 6502
;      ?RES  DEFL    0           ;;INDIREKTER SPRUNG ?
;      ;;
;      IRPC  X,(P1)
;      IF    '&X' EQ ''
;      ?RES  DEFL    NOT 0
;      ENDIF
;      EXITM
;      ENDM
;      ;;
;      IF    ?RES
;      DEFB  6CH           ;;INDIREKTER SPRUNG
;      .CREF
;      DEFW  P1
;      .XCREF
;      ELSE
;      DEFB  4CH           ;;ABSOLUTER SPRUNG
;      .CREF
;      DEFW  P1
;      .XCREF
;      ENDIF
;      ;;
;      ELSE
;      ;;
;      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
;      ?ER6502 DEFL  ?ER6502 + 1
;      ENDIF
;      ;;
;      ENDM

```

```

1
-----
;
; JSR: 6502 JSR-BEFEHL
;
+ .JSR MACRO P1
+ ;;
+ IF M6502 ;;ONLY IF 6502
+ DEFB 20H
+ .CREF
+ DEFW P1
+ .XCREF
+ ;;
+ ELSE
+ ;;
+ .PRINTX /-----> KEINE 6502-BEFEHLE !!/
+ ?ER6502 DEFL ?ER6502 + 1
+ ENDF
+ ;;
+ ENDM
;
-----
;
; LDA: 6502 LDA-BEFEHL
;
+ .LDA MACRO P1,P2
+ ;;
+ IF M6502 ;;ONLY IF 6502
+ ?CODE DEFL 0A1H
+ OPER1 P1,(P2)
+ ;;
+ ELSE
+ ;;
+ .PRINTX /-----> KEINE 6502-BEFEHLE !!/
+ ?ER6502 DEFL ?ER6502 + 1
+ ENDF
+ ;;
+ ENDM
;
-----
;
; LDX: 6502 LDX-BEFEHL
;
+ .LDX MACRO P1,P2
+ ;;
+ IF M6502 ;;ONLY IF 6502
+ ?CODE DEFL 0A6H
+ OPER2 P1,(P2),Y
+ ;;
+ ELSE
+ ;;
+ .PRINTX /-----> KEINE 6502-BEFEHLE !!/
+ ?ER6502 DEFL ?ER6502 + 1
+ ENDF
+ ;;
+ ENDM

```

```

;
;-----
;
; LDY: 6502 LDY-BEFEHL
;
; .LDY MACRO P1,P2
; ;;
; IF M6502 ;;ONLY IF 6502
; ?CODE DEFL 8A4H
; OPER2I P1,<P2>,X
; ;;
; ELSE
; ;;
; .PRINTX /-----> KEINE 6502-BEFEHLE !!/
; ?ER6502 DEFL ?ER6502 + 1
; ENDIF
; ;;
; ENDM

```

```

;
;-----
;
; LSR: 6502 LSR-BEFEHL
;
; .LSR MACRO P1,P2
; ;;
; IF M6502 ;;ONLY IF 6502
; ?CODE DEFL 846H
; OPER2A P1,P2
; ;;
; ELSE
; ;;
; .PRINTX /-----> KEINE 6502-BEFEHLE !!/
; ?ER6502 DEFL ?ER6502 + 1
; ENDIF
; ;;
; ENDM

```

```

;
;-----
;
; NOP: 6502 NOP-BEFEHL
;
; .NOP MACRO
; ;;
; IF M6502 ;;ONLY IF 6502
; DEFB 8EAH
; ;;
; ELSE
; ;;
; .PRINTX /-----> KEINE 6502-BEFEHLE !!/
; ?ER6502 DEFL ?ER6502 + 1
; ENDIF
; ;;
; ENDM

```



```

;
;-----
;
; PLA: 6502 PLA-BEFEHL
;
; .PLA MACRO
;
; IF M6502 ;;ONLY IF 6502
; DEFB 060H
;
; ELSE
;
;
; .PRINTX /-----> KEINE 6502-BEFEHLE !!/
; ?ER6502 DEFL ?ER6502 + 1
; ENDIF
;
; ENDM
;
;-----
;
; PLP: 6502 PLP-BEFEHL
;
; .PLP MACRO
;
; IF M6502 ;;ONLY IF 6502
; DEFB 20H
;
; ELSE
;
;
; .PRINTX /-----> KEINE 6502-BEFEHLE !!/
; ?ER6502 DEFL ?ER6502 + 1
; ENDIF
;
; ENDM
;
;-----
;
; ROL: 6502 ROL-BEFEHL
;
; .ROL MACRO P1,P2
;
; IF M6502 ;;ONLY IF 6502
; ?CODE DEFL 026H
; OPER2A P1,P2
;
; ELSE
;
;
; .PRINTX /-----> KEINE 6502-BEFEHLE !!/
; ?ER6502 DEFL ?ER6502 + 1
; ENDIF
;
; ENDM ENDF
;
; ENDM

```

```

-----
;
;
;      ROR:      6502 ROR-BEFEHL
;
+ .ROR  MACRO  P1,P2
+ ;;
+      IF      M6502  ;;ONLY IF 6502
+ ?CODE  DEFL  866H
+      OPER2A  P1,P2
+
+      ;;
+      ELSE
+      ;;
+      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
+ ?ER6502 DEFL  ?ER6502 + 1
+      ENDIF
+      ;;
+      ENDM

```

```

-----
;
;
;      RTI:      6502 RTI-BEFEHL
;
+ .RTI  MACRO
+ ;;
+      IF      M6502  ;;ONLY IF 6502
+      DEFB   40H
+
+      ;;
+      ELSE
+      ;;
+      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
+ ?ER6502 DEFL  ?ER6502 + 1
+      ENDIF
+      ;;
+      ENDM

```

```

-----
;
;
;      RTS:      6502 RTS-BEFEHL
;
+ .RTS  MACRO
+ ;;
+      IF      M6502  ;;ONLY IF 6502
+      DEFB   60H
+
+      ;;
+      ELSE
+      ;;
+      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
+ ?ER6502 DEFL  ?ER6502 + 1
+      ENDIF
+      ;;
+      ENDM

```





```

;
;-----
;
;      STX:   6502 STX-BEFEHL
;
;      .STX  MACRO  P1,P2
;      ;;
;      IF    M6502  ;;ONLY IF 6502
;      ?CODE DEFL  886H
;      OPER3  P1,P2,Y
;
;      ;;
;      ELSE
;
;      ;;
;      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
;      ?ER6502 DEFL ?ER6502 + 1
;      ENDIF
;
;      ;;
;      ENDM

```

```

;
;-----
;
;      STY:   6502 STY-BEFEHL
;
;      .STY  MACRO  P1,P2
;      ;;
;      IF    M6502  ;;ONLY IF 6502
;      ?CODE DEFL  884H
;      OPER3  P1,P2,X
;
;      ;;
;      ELSE
;
;      ;;
;      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
;      ?ER6502 DEFL ?ER6502 + 1
;      ENDIF
;
;      ;;
;      ENDM

```

```

;
;-----
;
;      TAX:   6502 TAX-BEFEHL
;
;      .TAX  MACRO
;      ;;
;      IF    M6502  ;;ONLY IF 6502
;      ?CODE DEFB  0AAH
;
;      ;;
;      ELSE
;
;      ;;
;      .PRINTX /-----> KEINE 6502-BEFEHLE !!/
;      ?ER6502 DEFL ?ER6502 + 1
;      ENDIF
;
;      ;;
;      ENDM

```



```

;-----;
; TXS: 6502 TXS-BEFEHL ;
;
; .TXS MACRO
; ;
; IF M6502 ;;ONLY IF 6502
; DEFB 89AH
; ;
; ELSE
; ;
; .PRINTX /-----> KEINE 6502-BEFEHLE !!/
; ?ER6502 DEFL ?ER6502 + 1
; ENDF
; ;
; ENDM
;-----;
;
; TYA: 6502 TYA-BEFEHL
;
; .TYA MACRO
; ;
; IF M6502 ;;ONLY IF 6502
; DEFB 89BH
; ;
; ELSE
; ;
; .PRINTX /-----> KEINE 6502-BEFEHLE !!/
; ?ER6502 DEFL ?ER6502 + 1
; ENDF
; ;
; ENDM
;-----;
;
; SUBTTL
; .CREF
; .LIST

```

## Macros:

.6502	.ADC	.ALLAB	.AND	.ASL	.BCC	.BCS	.BEG
.BIT	.BMI	.BNE	.BPL	.BRK	.BVC	.BVS	.CLC
.CLD	.CLI	.CLV	.CMP	.CPX	.CPY	.DEC	.DEX
.DEY	.EOR	.INC	.INX	.INY	.JMP	.JSR	.LDA
.LDX	.LDY	.LSR	.NO650	.NOP	.NORMA	.ORA	.PHA
.PHP	.PLA	.PLP	.ROL	.ROR	.RTI	.RTS	.SBC
.SEC	.SED	.SEI	.STA	.STX	.STY	.TAX	.TAY
OPER21	OPER3	OPER41	ZEROAB	?BRA	OPER1	OPER2	OPER2A

## Symbols:

?ER650 0000

No Fatal error(s)

\*\*\*\*\*  
Beispiel fuer die Anwendung des Makro-Pakets:

Der AP 22 BIOS Tastatur und 40 Zeichen Ausgabe Treiber.

Der AP 22 40 Zeichen Ein-/Ausgabe Treiber soll als Beispiel fuer die Verwendung des Makro-Pakets dienen, er stellt aber nur einen Teil des BIOS dar und ist allein nicht lauffaehig ...



AP4010 - APPLE 40 ZEICHEN EIN/AUSGABE

MACRO-88 3.4

01-Dec-80

PAGE

1

```

0000      CR      EQU      00H      ;WAGENRUECKLAUF ZEICHEN
      |
      |-----|
      |
      | VERWENDETE MONITOR-ROUTINEN
      |
      |
FBF4      ADVANCE EQU      0BF4H      ;CURSOR RECHTS
FDF0      COUT1  EQU      0FDF0H      ;CHARACTER OUTPUT
FC50      HOME   EQU      0FC50H      ;BILDSCHIRM LOESCHEN
FC42      CLREOP EQU      0FC42H      ;BIS BILDSCHIRMEDE
LOESCHEN
FC9C      CLREOL EQU      0FC9CH      ;BIS ZEILENOEDE LOESCHEN
FC22      UTAB   EQU      0FC22H      ;VERTIKAL ADR. NEU
BERECHNEN
FC24      UTAB2  EQU      0FC24H      ;BASIS ADR. ERRECHNEN
FC1A      UP     EQU      0FC1AH      ;CURSOR : ZEILE HOCH
FE04      SETNORM EQU      0FE04H      ;INITIALISIERUNG
FB2F      INIT   EQU      0FB2FH      ;DGL.
FE93      SETVID EQU      0FE93H      ;DGL.
FE09      SETK80 EQU      0FE09H      ;DGL.
      |
FF00      TASTST EQU      0FF00H      ;200 TASTATUR STATUS
0000      ZRUN   EQU      0
0000      IOSEL  EQU      0C000H

```

A40INI -

40 ZEICHEN EIN/AUSGABE INITIALISIEREN

```

0000'      A40INI:
      +      .LDA    #,0FFH
      +      .ALLABS
      +      .STA    LST1SLT      ;EINGABE SLOT RUECK.
      +      .STA    LST0SLT      ;AUSGABE SLOT RUECKS.
      +      .NORMAL
      +      .JSR    SETK80
      +      .JMP    SETVID

```

ABGEFRAGT

AP40STAT WIRD IM POLLING BETRIEB PERMANENT

ENTRY : Y - AP22 SLOT \* 16

Y DARF NICHT VERAEENDERT WERDEN !!

000E'

AP40STAT:

AP4010 - APPLE 48 ZEICHEN EIN/AUSGABE

MACRO-00 3.4

01-Dec-80

PAGE

1-3

```

+ .SEI ;NICHT UNTERBRECHBAR
+ .BIT KBD ;ZEICHEN BEREIT ?
+ .BPL NOTRDY
;
; ZEICHEN BEREIT
+ .LDA #,0FFH
+ .BNE RDY
;
0018' NOTRDY: .LDA #,0
;
001A' RDY:
+ .ALLABS ;ZERO PAGE ADRESS. ABSCHNITT
ALLEN
+ .CMP LAST
+ .BNE CHANGE
;
+ .CLI
+ .RTS
;
0021' CHANGE:
+ .STA LAST ;LETZTER = NEUER STATUS
+ .STA !0SEL+7,Y ;SELEKTIERE FF00 BEREICH
+ .STA TASTST ;STATUS ABSPEICHERN
+ .LDA !0SEL+2RUN+1,Y ;200 WIEDER STARTEN
+ .CLI
+ .RTS
;
SEN .NORMAL ;ZERO PAGE ADRESS. ZULASSE
002F' FF LAST: DEF0 0FFH ;LETZTER TASTATUR STATUS
;
; *****
;
; AP40 IN
;
; ! ZEICHEN VON DER APPLE-TASTATUR EINLESEN
;
; EXIT : PAR2H - ZEICHEN
;
-----
0030' AP40 IN:
+ .BIT KBD ;ZEICHEN BEREIT ?
+ .BPL AP40 IN ;NEIN ->
;
+ .LDA KBD ;ZEICHEN LESEN
+ .BIT KBDSTRB ;STATUS LOESCHEN
+ .AND #,7FH ;MSB LOESCHEN
+ .STA PAR2H
+ .RTS
;

```

```

;*****
;
; AP48OUT
;
; 48 ZEICHEN AUSGABE
;
; ENTRY : X - FUNKTIONNS-NR.
;         PAR3L - ZEICHEN (NICHT BEI FUNKTION < > )
;         POS1 - Y-POSITION (ABSOLUTE
POSITIONIERUNG)
;         POS2 - X-POSITION (ABSOLUTE
POSITIONIERUNG)
;
;-----
;
; AP48OUT:
;
; .ALLABS
; .LDA CURCHR ;UNTER-CURSOR ZEICHEN
; .LDY CHLOK ;CURSOR POSITION HORIZONT
TAL
;
; .STA 0,Y ;ZURUECKSPEICHERN
;
; POSL EQU 0-2 ;BEFEHL MODIFIZIEREN
; POSH EQU 0-1
;
; .TXA ;FUNKTIONNS-NR.
; .ASL A ;* 2
; .TAX ;IN X
; .LDA FUNCTB,X ;LSB(FUNKTIONNS-
UNTERPROGRAMM)
; .STA INDJMP ;ABSPEICHERN
; .LDA FUNCTB+1,X ;MSB(FUNKTIONNS
-UNTERPROGRAMM)
; .STA INDJMP+1 ;ABSPEICHERN
; .NORMAL
; .LDA PAR3L ;AUSGABE ZEICHEN IN A
; .JSR FUNCTION ;FUNKTIONNS-UP AUF EN
;
; .LDA BASL ;CURSOR POSITION RETTEN
; .ALLABS
; .STA POSL
; .NORMAL
; .LDA BASH
; .ALLABS
; .STA POSH
; .NORMAL
; .LDY CH
; .ALLABS
; .STY CHLOK
; .NORMAL
; .LDA <BASL>,Y ;ZEICHEN AN CURSOR-POS
; .ALLABS
; .STA CURCHR ;RETTE
; .NORMAL

```







AP4010 - APPLE 40 ZEICHEN EIN/AUSGABE    MACRO-80 3.4    01-Dec-80    PAGE 5

## Macros:

.5502	.ADC	.ALLAB	.AND	.ASL	.BCC	.BCS	.BEQ
.BIT	.BMI	.BNE	.BPL	.BRK	.BVC	.BVS	.CLC
.CLD	.CLI	.CLV	.CMP	.CPX	.CPY	.DEC	.DEX
.DEY	.EDR	.INC	.INX	.INY	.JMP	.JSR	.LDA
.LDX	.LDY	.LSR	.NO650	.NOP	.NORMA	.ORA	.PHA
.PHP	.PLA	.PLP	.ROL	.ROR	.RTI	.RTS	.SBC
.SEC	.SED	.SEI	.STA	.STX	.STY	.TAX	.TAY
.TSX	.TXA	.TXS	.TYA	?BRA	OPER1	OPER2	OPER2A
OPER21	OPER3	OPER41	ZER0AB				

## Symbols:

..0000	FFFF	..0001	FFFF	..0002	FFFF	..0003	0000
..0004	0000	..0005	FFFF	..0006	FFFF	..0007	FFFF
..0008	FFFF	..0009	FFFF	..000A	0000	..000B	0000
..000C	0000	..000D	0000	..000E	0000	..000F	0000
..0010	FFFF	..0011	FFFF	..0012	FFFF	..0013	FFFF
..0014	FFFF	..0015	FFFF	..0016	0000	..0017	0000
..0018	FFFF	..0019	FFFF	..001A	FFFF	..001B	FFFF
..001C	FFFF	..001D	FFFF	..001E	FFFF	..001F	FFFF
..0020	FFFF	..0021	FFFF	..0022	FFFF	..0023	FFFF
..0024	FFFF	?ALLAB	0000	?CODE	0001	?ER650	0000
?RES	0000	A401N1	00001'	ADWANC	FBF4	AP401N	00301'
AP400U	00401'	AP405T	000E1'	BASH	0029	BASL	0020
CH	0024	CHANGE	0021'	CHLOK	0005'	CKREST	000A'
CLRE0L	FC9C	CLRE0P	FC42	COUT1	FD00	CR	0000
CURCHR	0006'	CJ	0025	FUNC0	0001'	FUNC4	0096'
FUNC5	0099'	FUNC6	009E'	FUNC7	00A9'	FUNCTB	00C1'
FUNCT1	007E'	HOME	FC50	INDJMP	00D7'	INIT	FB2F
INVFLG	0032	IOSEL	C000	KBD	C000	KBDSTR	C010
LAST	002F'	LSTISL	0003*	LSTOSL	0006*	M6502	FFFF
MASKE	000F*	NOBU	0077'	NOBUCH	0091'	NOTRDY	0018'
PAR2H	00FD	PAR3L	00FE	POS1	0006*	POS1ER	000E'
POS2	00AA*	POS2ER	0002'	POSH	0040'	POSJ	0047'
RDY	001A'	SETKBD	FE09	SETNOR	FE04	SETVID	FE93
TASTST	FF00	UP	FC1A	VTAB	FC22	VTAB2	FC24
WNB0TM	0023	WNDLFT	0020	WNDTOP	0022	WNDW0T	0021
ZRUN	0000						

No Fatal error(s)

## Anhang G

## Belegung der BOOT-Spuren

In diesem Abschnitt finden Sie die Belegung der Spuren 0 bis 2 von kaltstartfähigen Apple II Disketten. Um diese Spuren schnell laden zu können werden sie mit Skew-Faktor 2 gelesen, d. h. zwischen 2 aufeinander folgenden Sektoren ist genau 1 Sektor frei. Neben den Sektoren ist jeweils die zugehörige Adresse des 256 Byte Bereichs in dem sich der Inhalt am Ende der Kaltstart Phase befindet, und die Adresse in der Datei IBSCPM64.COM angegeben (wenn sie mit einem Debugger ohne Offset geladen wird). In der Spalte 'Bedeutung' ist angegeben wofür der jeweilige Bereich im IBS CP/M System benutzt wird. Adressen die mit \$ anfangen befinden sich im Apple II Speicher, werden sie mit H beendet im AP 22 Speicher.

Spur	Sektor	Zieladresse	IBSCPM64 Adresse	Bedeutung
0	0	\$0800 - \$08FF	0800H - 08FFH	BOOT0 Lader
0	1	\$1000 - \$10FF	1000H - 10FFH	BIOS 6502 Teil
0	2	\$0900 - \$09FF	0900H - 09FFH	BOOT1 Lader
0	3	\$1100 - \$11FF	1100H - 11FFH	BIOS 6502 Teil
0	4	\$0A00 - \$0AFF	0A00H - 0AFFH	BOOT1 Lader
0	5	\$1200 - \$12FF	1200H - 12FFH	BIOS 6502 Teil
0	6	FF00H - FFFFH	0B00H - 0BFFH	Seite FF / AP22
0	7	\$1300 - \$13FF	1300H - 13FFH	BIOS 6502 Teil
0	8	\$0C00 - \$0CFF	0C00H - 0CFFH	BIOS 6502 Teil
0	9	\$1400 - \$14FF	1400H - 14FFH	BIOS 6502 Teil
0	10	\$0D00 - \$0DFF	0D00H - 0DFFH	BIOS 6502 Teil
0	11	\$1500 - \$15FF	1500H - 15FFH	BIOS 6502 Teil
0	12	\$0E00 - \$0EFF	0E00H - 0EFFH	BIOS 6502 Teil
0	13	\$1600 - \$16FF	1600H - 16FFH	BIOS 6502 Teil
0	14	\$0F00 - \$0FFF	0F00H - 0FFFH	BIOS 6502 Teil
0	15	\$1700 - \$17FF	1700H - 17FFH	BIOS 6502 Teil

Spur	Sektor	Zieladresse	IBSCPM64 Adresse	Bedeutung
1	0	\$1800 - \$18FF	1800H - 18FFH	BIOS 6502 Teil
1	1	E400H - E4FFH	2000H - 20FFH	CCP + 0400H
1	2	\$1900 - \$19FF	1900H - 19FFH	BIOS 6502 Teil
1	3	E500H - E5FFH	2100H - 21FFH	CCP + 0500H
1	4	\$1A00 - \$1AFF	1A00H - 1AFFH	BIOS 6502 Teil
1	5	E600H - E6FFH	2200H - 22FFH	CCP + 0600H
1	6	0000H - 00FFH	1800H - 18FFH	BIOS 6502 Teil
1	7	E700H - E7FFH	2300H - 23FFH	CCP + 0700H
1	8	E800H - E8FFH	1C00H - 1CFFH	CCP + 0800H
1	9	E900H - E9FFH	2400H - 24FFH	BDOS + 0000H
1	10	E100H - E1FFH	1000H - 10FFH	CCP + 0100H
1	11	E900H - E9FFH	2500H - 25FFH	BDOS + 0100H
1	12	E200H - E2FFH	1E00H - 1EFFH	CCP + 0200H
1	13	EA00H - EAFFH	2600H - 26FFH	BDOS + 0200H
1	14	E300H - E3FFH	1F00H - 1FFFH	CCP + 0300H
1	15	EB00H - EBFFH	2700H - 27FFH	BDOS + 0300H

Spur	Sektor	Zieladresse	ISDCPM64 Adresse	Bedeutung
2	0	EC00H - ECFFH	2800H - 28FFH	800S + 0400H
2	1	F400H - F4FFH	3000H - 30FFH	800S + 0C00H
2	2	ED00H - EDFFH	2900H - 29FFH	800S + 0500H
2	3	F500H - F5FFH	3100H - 31FFH	800S + 0D00H
2	4	EE00H - EEFH	2A00H - 2AFFH	800S + 0600H
2	5	F600H - F6FFH	3200H - 32FFH	BIOS + 0000H
2	6	EF00H - EFFFH	2B00H - 2BFFH	800S + 0700H
2	7	F700H - F7FFH	3300H - 33FFH	BIOS + 0100H
2	8	F800H - F8FFH	2C00H - 2CFFH	800S + 0800H
2	9	F800H - F8FFH	3400H - 34FFH	BIOS + 0200H
2	10	F100H - F1FFH	2D00H - 2DFFH	800S + 0900H
2	11	F900H - F9FFH	3500H - 35FFH	BIOS + 0300H
2	12	F200H - F2FFH	2E00H - 2EFFH	800S + 0A00H
2	13	FA00H - FAFFH	3600H - 36FFH	BIOS + 0400H
2	14	F300H - F3FFH	2F00H - 2FFFH	800S + 0B00H
2	15	FB00H - FBFFH	3700H - 37FFH	BIOS + 0500H

