



Monitor für die c't180-Karte

Start ohne CP/M

Andreas Burgwitz

Der beste Computer nutzt ohne Software wenig – der c't180 macht da keine Ausnahme. Eine 'Mindestmenge' Software ist schon notwendig, damit der Rechner nach dem Einschalten seine Meldung auf den Bildschirm schreibt. Diese Mindestmenge Software und noch etwas mehr ist beim c't180 in einem EPROM untergebracht: Routinen für die Zeichen-Ein-/Ausgabe, Test-Programme für die Hardware und ein Monitor-Programm. Ausgaben des Monitors nach Reset und der Eingabe von '?'

Dieses Monitor-Programm kann natürlich nicht ein Betriebssystem wie CP/M ersetzen, obwohl es fast alle Funktionen eines 'richtigen' Betriebssystems bietet. Der Monitor soll vielmehr den c't180 nach dem Einschalten in einen definierten Zustand bringen und bei der Inbetriebnahme sowie bei einer eventuell nötigen Fehlersuche dem Anwender helfen.

Initialisieren

Nach einem Reset (oder Sprung nach Adresse 0) muß das Programm die PIO (IC11) initialisieren, ein Konfigurations-Byte von der PIO lesen und zum Monitor-Start oder zu einer Test-Routine verzweigen. Die Init-Routine überprüft die Funktion der PIO, indem an den Port A der Wert AAh ausgegeben und wieder zurückgelesen wird. Tritt dabei ein Fehler auf, läuft die Routine auf einen Halt-Befehl. Andernfalls prüft das Programm, ob ein Test ge-

wählt wurde. Ist das nicht der Fall, verzweigt das Programm zu den Initialisierungsroutinen des Monitors, andernfalls zu der geforderten Test-Routine.

Bei der Initialisierung des Monitors werden die Adresse der PIO sowie das Config-Byte im RAM abgelegt. Das Config-Byte enthält Informationen über die Quarz-Frequenz, das Übertragungsformat CPU/Terminal, die Lage der CPU-internen I/O-Page, das Boot-Format sowie Angaben, welcher Test durchgeführt werden soll.

Anschließend initialisiert das Programm die SIOs, wählt die Routinen für Zeichen-Ein-/Ausgabe aus und kopiert sie ins RAM. Es folgt das Löschen der Monitor-Pufferspeicher und die Ausgabe der Meldungen: Monitor-Version, CPU-I/O-Page, PIO-Adresse, Quarzfrequenz, erkannte Floppy-Karte sowie das gewählte Floppy-Format.

Die HD64180-CPU kennt wie der Z80-Prozessor Restart-Befehle. Die Restart-Adressen 8h, 10h, 18h, 20h, 28h, 30h sowie die NMI-Adresse 66h führen auf den Start des EPROM-Programms, also auf die Adresse 0. Die Restart-Adresse 38h führt jedoch auf eine Breakpoint-Routine.

Nach seiner Initialisierung und der Ausgabe der Meldungen wartet der Monitor auf die Eingabe eines Befehls. Eine Anweisung besteht immer aus einem Buchstaben und bis zu drei Argumenten. Der Buchstabe kennzeichnet die gewünschte Moni-

Bedienung

tor-Funktion und muß direkt nach dem Prompt-Zeichen (#) des Monitors eingegeben werden. Die einzelnen Argumente sind untereinander und von dem Befehlsbuchstaben durch ein Leerzeichen (Blank) zu trennen. Während der Eingabe kann man mit 'Backspace' Korrekturen vornehmen. Den Abschluß der Eingabe kennzeichnet man mit RETURN. MONI2 interpretiert alle Argumente als Zahlen im hexadezimalen Zahlensystem. Führende Nullen müssen nicht eingegeben werden. MONI2 akzeptiert die Eingabe von Klein- oder Großbuchstaben.

MONI2 geht davon aus, daß an der CPU-Karte entweder die c't86-Floppy-Karte oder die Intelligente Floppy-Karte (IFC) betrieben wird. Die Unterscheidung zwischen den Karten nimmt der Monitor automatisch vor und verzweigt bei einem Floppy-Zugriff zu den jeweils relevanten Treiber-Routinen. Hat man keine dieser Floppy-Karten oder eine MONI2 nicht bekannte Karte mit dem System verbunden, erfolgt bei dem Versuch eines Floppy-Zugriffs eine Fehlermeldung. Alle Zugriffe auf ein Floppy-Laufwerk beziehen sich auf Laufwerk 'A'.

Der Monitor kennt die im folgenden aufgeführten Anweisungen.

A (Arithmetik)

Addiert den als Argument 1 (ARG1) angegebenen Wert mit dem von Argument 2 (ARG2). Nach der Ausgabe der Summe führt die Routine die Subtraktion ARG1 – ARG2 durch.

B (Breakpoint)

An die Speicherstelle ARG1 wird ein Software-Haltpunkt gesetzt. Der Breakpoint (0FFh – RST 38h) muß immer am Anfang eines Maschinenbefehls stehen. Beim Setzen des Breakpoints wird der Inhalt der Speicherzelle gegen den Restart-Befehl getauscht. Der alte Speicherinhalt wird in einem Pufferspeicher gesichert. Mit dem Aufruf von 'B' ohne Angabe ei-

nes Arguments wird der ursprüngliche Speicherinhalt wieder zurückgeladen und der Haltpunkt gelöscht.

Trifft der Monitor bei der Ausführung eines Programms auf einen Breakpoint, gibt er den Inhalt der Register AF, BC, DE, HL, PC, SP sowie den Inhalt der Speicherzelle, die indirekt durch dieses Register adressiert ist auf den Bildschirm aus. Das Flag-Register des Akkus wird in binärer Darstellung mit der Bezeichnung der Flags ausgegeben.

C (Copy Speicher)

Kopiert den Speicherinhalt ab der Adresse ARG1 bis Adresse ARG2 in den Speicherbereich ab Adresse ARG3. Dabei darf ARG1 nicht größer als ARG2 sein.

F (Fill Speicher)

Füllt den Speicher ab der Adresse ARG1 bis Adresse ARG2 mit dem Byte, das in ARG3 angegeben ist.

H (Dump Speicher)

Zeigt den Speicher ab Adresse ARG1 bis Adresse ARG2 in hexadezimaler und ASCII-Darstellung auf dem Bildschirm an. Gibt man nur eine Start-Adresse an, also nur ARG1, zeigt MONI2 den Speicherbereich von ARG1 bis ARG1 + 80h an. Die Ausgabe kann durch Eingabe von 'S' (Control S) angehalten und durch erneute Eingabe von 'S' wieder gestartet werden. Die Eingabe von '.' beendet die Befehlsausführung.

I (In Port)

Liest die in ARG1 angegebene Port-Adresse und zeigt den gelesenen Wert in binärer und hexadezimaler Darstellung auf dem Bildschirm an. Gibt man als zweites Argument den Buchstaben 'C' (Continue) ein, wird der Port einmal gelesen, der Wert ausgegeben und anschließend der Port so lange abgefragt, bis eine Änderung des Wertes eintritt. In diesem Fall wird der neue Wert ausgegeben und wieder zur Befehlseingabe verzweigt. Die Warteschleife kann man jederzeit durch Eingabe von '.' verlassen.

J (Jump Adresse)

Der Programm-Counter der CPU wird auf die in ARG1 angegebene Adresse gesetzt. Das entspricht einem Sprung zu der in ARG1 angegebenen Adresse.

M (Modify Speicher)

Bietet die Speicherzelle ARG1

zur Modifikation an. Mit der Eingabe von Return bietet MONI2 die nächste Speicherzelle zur Korrektur an, mit der Eingabe von '.' die vorstehende Zelle. Dabei bleibt der Inhalt des Speichers unverändert. Gibt man einen Wert an (ein Byte, also zwei Zeichen), wird dieser nach Eingabe von Return in den Speicher übernommen und die nächste Speicherzelle zur Korrektur angeboten. Alle Eingaben werden als hexadezimale Zahlen interpretiert. Für die direkte Eingabe von ASCII-Zeichen muß vor dem Zeichen ein Hochkomma stehen (Beispiel 'A < CR >'). MONI2 übernimmt in diesem Fall den ASCII-Code des Zeichens in den Speicher. Durch die Eingabe eines Punktes kann man den Modify-Modus verlassen und wieder zur Befehlsebene des Monitors gelangen.

^P (Printer Toggle)

Die Eingabe von ^P (Control P) schaltet einen Drucker 'parallel' zum Bildschirm. Die wiederholte Eingabe von ^P schaltet den Drucker wieder aus. Jedes Zeichen, das der Computer an den Bildschirm sendet, gelangt auch an den Drucker.

R (Read Sektor)

Liest einen Sektor von der Floppy. Die Daten werden in dem Speicherbereich ab der Adresse ARG1 abgelegt.

S (Set Sektor)

Bei einem folgenden Floppy-Zugriff wird der in ARG1 angegebene Sektor gelesen oder beschrieben.

Befehl	Argumente
A Arithmetik	Wert 1 Wert 2
B Breakpoint	Break-Adresse
C Copy Speicher	Start Ende Ziel
F Fill Speicher	Start Ende Füllbyte
H Dump Speicher	Start Ende
I In Port	Port-Adresse
J Jump Adresse	Adresse der User-Routine
M Modify Speicher	Startadresse
O Out Port	Port-Adresse Ausgabebyte
R Read Sektor	Ziel-Adresse
S Set Sektor	Sektornummer
T Set Track	Spurnummer
V Verify Speicher	Start Ende Ziel
W Write Sektor	Ziel-Adresse
X Keine Funktion	--
Z Boot Betriebssystem	--
? Hilfs-Menü	--

Diese Befehle kennt MONI2

T (Set Track)

Der Kopf des Floppy-Laufwerks wird über die in ARG1 angegebene Spur gestellt.

V (Verify Speicher)

Vergleicht den Speicherinhalt ab Adresse ARG1 bis Adresse ARG2 mit dem Speicher ab der Adresse ARG3. Sind die Werte der Speicherzellen gleich, erfolgt keine Ausgabe. Andernfalls zeigt MONI2 die beiden Adressen und den Inhalt beider Speicherzellen auf dem Bildschirm an. Durch die Eingabe eines Punktes gelangt man vor dem Ende der Befehlsausführung in die Befehlsebene.

W (Write Sektor)

Schreibt einen Sektor auf die Floppy. Die Daten ab der Speicherstelle ARG1 werden auf die Diskette geschrieben. Dabei gelten die Spur und der Sektor, die zuvor mit den Befehlen 'T' und 'S' eingestellt wurden.

Z (Boot Betriebssystem)

Liest den ersten Sektor der ersten Spur von Diskette und lädt die Daten ab der Adresse 9000h in den Speicher. Nach dem Ladevorgang wird die Befehlsausführung an der Adresse 9000h fortgesetzt, also das geladene Programm ausgeführt.

? (Hilfe)

Eine Tabelle mit den zur Verfügung stehenden Befehlen des Monitors sowie der Angabe der notwendigen Argumente wird auf den Bildschirm ausgegeben.

In der Kürze...

MONI2 bietet einige Möglichkeiten, die Eingabe der Befehle

zu verkürzen. Gibt man RETURN ein (ohne Befehlsbuchstaben und Argumente), wiederholt MONI2 den zuletzt ausgeführten Befehl. Dabei ver-

HD64180-Monitor Version: 1.0
01.86
A.Burgwitz, R.Kress

Konfiguration:
Interne I/O-Page = 40
PIO Basis-Adresse = 00
Quarz-Frequenz = 12,288 MHz
c't-Floppy-Karte
5,25", SS, DD, 1024 Byte/Sektor

#?	Befehl	Argumente
A	Arithmetik	WERT1 WERT2
B	Breakpoint	Adresse
C	Copy Mem.	VON BIS NACH
F	Fill Mem.	VON BIS WERT
H	Dump Mem.	VON BIS
I	In Port	PORT-ADRESSE
J	Jump	ADRESSE
M	Modify Mem.	ADRESSE
O	Out Port	PORT-ADRESSE WERT
^P	Drucker Ein/Aus	
R	Read Sektor	ZIELADRESSE
S	Set Sektor	SEKTOR-NUMMER
T	Set Track	SPUR-NUMMER
V	Verify Mem.	VON BIS MIT
W	Write Sektor	STARTADRESSE
X	User-Routine	
Z	Boot Betriebssystem	
?	Hilfs-Menue	
	RS38h führt auf eine Breakpoint-Routine	

Ausgaben des Monitors nach Reset und der Eingabe von '?'

wendet der Monitor das End-Argument als neues Start-Argument. Hatte man zum Beispiel den Speicherbereich von 0100h bis 0130h aufgelistet und gibt anschließend nur RETURN ein, listet der Monitor den Speicherbereich von 0130h bis 01B00h.

Eine weitere Möglichkeit der Befehlsverkürzung besteht darin, nur einen Befehlsbuchstaben ohne Argumente anzugeben. MONI2 verwendet wiederum das End-Argument als Start-Argument für den neuen Befehl. Zum Beispiel nach einem Dump des Speichers von 0100h bis 0130h bietet der Monitor nach der Eingabe von 'M < CR >' die Speicherzelle 0130h zur Korrektur an.

Expansion

Für eigene Erweiterungen von MONI2 ist die Funktion X vorgesehen. Bei der Eingabe dieses Buchstabens verzweigt MONI2 auf das letzte Byte, das vom Monitor-Programm belegt ist. Hier steht üblicherweise ein Return-Befehl, so daß wieder zur Befehlseingabe verzweigt wird. An diese Stelle im EPROM kann man jedoch sehr einfach eine eigene Routine setzen.