SD-Card

Betrieb einer SD-Card unter CP/M mit minimaler Hardware

Hier ist er endlich - mein 1-Euro-SD/MMC-Card-Anschluss!

Mein Modul sollte so einfach wie möglich sein und so wenig wie möglich kosten. Bekannte Lösungen enthalten einen zusätzlichen Mikrocontroller (PIC oder AVR) als Vorraussetzung für ein FAT16-Filesystem, damit ist so ein Zusatzmodul eigentlich schon leistungsfähiger als der KC selbst, s. sdcard2.

Letztlich enthalten die MMC- und SD-Karten schon einen eigenen Mikrocontroller, der den Zugriff auf den Kartenspeicher recht einfach macht. Meine Idee war es, die MMC/SD-Karten nur unter CP/M anzusprechen und dessen Filesystem zu nutzen. Damit wird der zusätzliche Mikrocontroller überflüssig.

Ich hatte im Netz bei Ulrich Radig eine einfache Schaltung mit 6 Widerständen und 2 Dioden gefunden (herabsetzen der 5V auf die erlaubten 3,3V); bei einem Z80F91 -Projekt (Jean-Michel Howland) gab es sogar ein komplettes CP/M mit SD-Einbindung über die im Z80F91 vorhandene serielle Schnittstelle.

Hier ein paar mehr oder weniger bekannte Links zum Thema MMC/SD und Z80:

- http://elm-chan.org/docs/mmc/mmc_e.html
- http://hotbit.blogspot.com/search/label/Leitor%20de%20SD/MMC
- http://www.vegeneering.com/eZ80_CPM/index.html
- http://n8vem-sbc.pbwiki.com/browse/#view=ViewFolder&param=Juha%20SD%20Interface

Es gibt eine Reihe von Implementierungen des FAT-Filesystems für die AVR-Controller, z.B. Holger Klabundes Variante. Zuerst plante ich ein spezielles Modul zum Einstecken in den Z9001. Den Userport wollte ich für Soundmodul etc. frei halten. Neben dem AVR hatte ich an einen EPROM mit dem Steuerprogramm gedacht (2008: Andere hatten auch diese Idee und das viel besser gelöst → sdcard2). Dann bin ich später auf den Trichter mit der direkten Hardwareansteuerung gekommen.

Schaltung

Die Minimal-Schaltung von Ulrich Radig. Die Widerstände und die Dioden sind nötig, da SD-Karten mit 3,3V arbeiten. Die Schaltung ist wirklich minimal; man sollte besser einen Spannungsregler einsetzen.

Der Aufbau erfolgte auf einer Lochrasterplatte 30×45 mm und mit Handverdrahtung (Verowire Fädelstift) und funktionierte auf Anhieb. Gegenüber der Schaltung habe ich drei SAY12-Dioden verwendet, um die Betriebsspannung auf 3,6 V zu senken. Die Ports können beliebig auf die Anschlüsse des USER-Port gelegt werden, die Pins können einfach im Quellcode umdefiniert werden. Ich habe bei meinem Muster die Reihenfolge wie im Bild angegeben übernommen. Als Halterung für die Karte dient ein Stück ISA-Karten-Slot. **Achtung: Die Karte darf nicht im laufenden Betrieb gewechselt werden!!!**

Wer es richtig professionell haben will, findet z.B. bei DISPLAY 3000 eine Lösung. Auf dieser Seite gibt es auch ausführliche Erklärungen, warum die Pegelwandlung mit Widerständen nur als Bastellösung anzusehen ist.

Software

Die Karte wird direkt im SPI-Modus angesteuert. Das ganze ist relativ simpel, da der Master (hier der Z9001) den Takt beliebig vorgeben kann. In Rahmen eines Seminars „Machbarkeitsstudie: Einchipcomputer“ an der Uni Koblenz hat Simeon Maxein dieses Vorgehen einsteigerfreundlich beschrieben http://www.uni-koblenz.de/~physik/informatik/ECC/sd.pdf, http://www.uni-koblenz.de/~physik/informatik/ECC/.

Den Zugriff auf die MMC bzw. SD erledigt der Dateitreiber von **CP/M**; eine Kompatibilität zu FAT16 ist dabei nicht nötig. Emuliert wird dann auf der Karte eine/zwei CP/M-Festplatten bis 8MB. Wenn diese nicht auf Block 0 auf der MMC anfangen, sondern weiter hinten, kann man hier jedoch sogar noch einen FAT16-Header mit draufschreiben und die CP/M-„Festplatten“ direkt am PC und am Stück kopieren: Unter DOS erscheinen 2 8MB große Dateien im Directory!


Auflisten des BOOT-Sektors der SD-Karte

Dazu werden im BIOS anstelle des Floppytreibers (oder zusätzlich zu diesem) Treiberroutinen eingebunden, die auf den Quellen von Jean-Michel Howland basieren und die 128 Byte-Records direkt auf die Karte referenzieren. Damit kann mit der SD-Karte wie mit einer richtigen „großen“ Floppy bzw. Festplatte gearbeitet werden. Am ehesten ist das mit einer RAM-Disk vergleichbar. Das Lesen und Schreiben geht leider noch so schnell, dafür rattert im Gegensatz zur Floppy auch nichts 🙂

27.02.2010: Der aktuelle Treiber SDC2 sdc2.zip wird wie der RAM-Floppy Treiber für die RAF2008 nachgeladen und installiert sich als Laufwerk 'N:'. Der Treiber funktioniert, ist aber noch nicht optimal: Das Lesen oder Schreiben von 1K dauert knapp 1 Sekunde (das Lesen eines Bytes erfordert allein 17 In/Out-Befehle. Daraus resultiert die langsame Transferrate). Die Karte wird mit 512 Byte-Blöcken gelesen und beschrieben, davon werden aber nur 128 Byte genutzt. Für ein 8MB-Laufwerk braucht man damit eine 32MByte-SD-Karte. Der nächste Treiber wird einen Deblocking-Mechanismus enthalten, so dass zum einen auch echte 8MB ausreichen und zum anderen die Transferrate vervierfacht wird.

Im aktuellen Treiber sind übrigens 2 Tracks (64 Sektoren) für Systemspuren reserviert. Wenn man eine kleine SD-Karte mit FAT formatiert, steht in Track 0 der FAT-Disk-Header („MSDOS5.0“) und in Track 1 das FAT-Inhaltsversichnis. Auf Track 2 beginnt die erste Datei. Somit reicht es aus, die im

Archiv enthaltene Datei „TEST.CPM“ auf die frisch formatierte SD-Karte zu kopieren und schon hat man das CP/M-Filesystem auf der Karte, verpackt in der virtuellen MSDOS-Datei „TEST.CPM“:



[] [] []

In Andrew Lynchs N8VEM-Projekt gibt es ebenfalls eine Nutzung einer SD-Karte im CPM incl Unterlagen, das Juha SD Interface (s.o.).

From:
https://hc-ddr.hucki.net/wiki/ - **Homecomputer DDR**

Permanent link:
https://hc-ddr.hucki.net/wiki/doku.php/z9001/cpm/sdcard?rev=1279882279

Last update: **2010/07/22 22:00**