SD / MMC – Karten Bootloader Dokumentation (V1.0)

Features

Ich habe mir zum Ziel gesetzt einen Bootloader für Atmels Atmega Controller zu schreiben, der sich die neue Firmware jeweils von einer SD bzw. MMC Karte holt. Dabei herausgekommen ist ein Programm mit folgenden Eigenschaften:

- Unterstützung von SD / MMC Karten
- FAT32 als Filesystem (FAT16 wird u.U. folgen)
- passt in eine 1024 Word (2048Byte) Bootsection, läuft somit auch auf Controllern ab 8kb Flash
- liest binary Files (keine intelhex!!)
- Getestet auf einem Atmega168(8mhz RC-OSC), Speichermedium : 512mb Sandisk SD, compiliert mit AVR-GCC 3.4.6

Einstellungen, Compilieren, Fusebits

Alle relevanten Einstellungen können in config.h getätigt werden. Wichtig sind vor allem die Einstellungen für die SPI-Pins, damit diese korrekt konfiguriert werden können.

Zusätzlich muss im makefile festgelegt werden, welchen Controller man verwendet und an welche Adresse der Bootloader gelinkt werden soll. Die entsprechende Stelle im Datenblatt findet man bei „Bootloadersupport", im Datenblatt des Atmega8 zum Beispiel auf Seite 217.
Den Wert für die „Boot Reset Adress" bei einer Bootsize von 1024 Words, in diesem Falle 0xC00 multipliziert man mit 2 (Umrechnung Wordadresse in eine Byteadresse). Für einen Atmega8 wäre der Wert somit 0x1800. Die entsprechende Zeile im Makefile müsste also wie folgt lauten:

```
BTLOADER_ADRESS = 0x1800
```

In seltenen Fällen muss eventuell die Geschwindigkeit des SPI Busses angepasst werden, falls die Karte nicht reagiert.

Das Compilieren sollte eigentlich ohne Probleme verlaufen, der GCC 3.4.6 gibt mir allerdings eine Warnung aus, deren Ursache ich nicht kenne. Vielleicht handelt es sich dabei aber auch nur um ein zu altes Def-File des Controllers den ich verwende(ATmega168)

```
#warning "The functions from <avr/eeprom.h> are not supported on this MCU."
```

Die Fusebits müssen so gesetzt werden, dass der Controller eine Bootsize von 1024 Words verwendet (BOOTSZ Fuses) und der Resetvektor ein Bootloaderreset auslöst. (BOOTRST Fuse)

Wichtig!

- Der Bootloader unterstützt bisher nur binary files und kein IntelHex Format! Bei AVRGCC lässt sich per makefile auf einfach Weise einstellen, ob eine ihex oder binary Datei erstellt werden soll. Im Beispielprojekt ist bereits ein dementsprechend angepasstes makefile vorhanden.
- Die Datei auf der Karte muss „MAIN.BIN" genannt werden, damit der Bootloader sie als Firmwareupdate erkennt.

To Do

- Modified-Time-Check, damit das Firmwareupdate nur erfolgt wenn auch eine aktuellere Version auf der Karte gefunden wird (schont Flash)
- FAT16 Unterstützung
- bessere ‚Reset' Routine (gemeint ist der Sprung zur Adresse 0, Applikationsstart)
- eventuell IntelHex Unterstützung(passt aber kaum mehr in 2kb Bootsection)

Lizenz

Changelog

V1.0 - 5.11.2006

Binary Files, FAT32, tested on Atmega168@8Mhz, 512mb Sandisk SD, compiled with AVR-GCC 3.4.6

Autor

Nik Bamert
nikbamert@bluewin.ch
www.nikbamert.com