FHWS
Fachhochschule Würzburg-Schweinfurt
University of Applied Sciences

In Zusammenarbeit mit

cor science

# Diplomarbeit

## Konzeption einer Entwicklungsplattform für 'Embedded Linux' auf Basis der ARM9 Technologie

von
**Andreas Bießmann**
20. Januar 2008

| | |
|---|---|
| **Erstprüfer:** | Prof. Dr.-Ing. Ludwig Eckert |
| **Zweitprüfer:** | Prof. Dr.-Ing. Martin Ochs |
| **Firmenbetreuer:** | Dr.-Ing. Claudius Moor |

## Zusammenfassung

Diese Diplomarbeit beschreibt ein Entwicklungssystem für Embedded Linux auf Basis eines ARM9 Rechnersystems.

Es soll einem Anwendungsentwickler auf einfache Art und Weise die Möglichkeit geben, Embedded Linux Anwendungen für ein ARM9 System zu entwickeln.

Dazu beschreibt diese Arbeit die Umsetzung einer Hardwarespezifikation, die einzelnen Schritte zum Erstellen der Firmware sowie das Zusammenspiel der Werkzeuge auf einem Entwicklungsrechner mit den Programmen auf der Testhardware.

# Danksagung

Ich möchte mich bei meinem betreuenden Professor Dr. Eckert für zahlreiche Hinweise, Literaturangaben und Denkanstöße bedanken.

Bei der Firma Corscience und allen Mitarbeitern möchte ich mich für die gute Unterstützungen bedanken. Im Speziellen möchte ich einige Personen besonders hervorheben.

Bei Norman Jörns möchte ich mich für das anregende Gespräch über das Thema Embedded Linux vor dem Beginn dieser Diplomarbeit bedanken. Dieses war für mich der ausschlaggebende Moment, mich für das Betriebssystem Linux auf einem Embedded System zu faszinieren.

Bei Thomas Weber und Manuel Seufert möchte ich für das Korrekturlesen dieser Arbeit und für die vielen hilfreichen Bemerkungen bedanken.

Ein besonderer Dank gilt meiner Freundin Kristina Pusch, die während dem Entstehen dieser Arbeit viel Verständniss für die zeitraubende Schreibarbeit zeigte.

Und zu guter Letzt möchte ich mich bei meinen lieben Eltern für die großartige Unterstützung seit meiner Entscheidung, ein Studium auf dem zweiten Bildungsweg durchzuführen, bedanken. Ohne Euch hätte ich mich nicht in der Lage gesehen, diese Anstrengung zu meistern.

## Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbstständig verfasst und noch nicht anderweitig für Prüfungszwecke vorgelegt habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und habe wörtliche oder sinngemäße Zitate als solche gekennzeichnet.

Erlangen, den 20. Januar 2008

.....

*Andreas Bießmann*

# Inhaltsverzeichnis

**Abkürzungsverzeichnis** v

**Abbildungsverzeichnis** ix

**Tabellenverzeichnis** xi

**Listings** xiii

**1 Einleitung** 1

1.1 Problemstellung . . . . . 2

1.2 Anforderungen des Entwicklungssystems . . . . . 4

**2 Einführung in die Grundsätze freier Software** 7

2.1 Lizenzmodelle freier Software . . . . . 9

2.1.1 GNU General Public License (GPL) . . . . . 10

2.1.2 GNU Lesser General Public License (LGPL) . . . . . 11

2.2 Entstehung des Linux Kernels . . . . . 12

2.3 Kommerzielle Anwendungen von freier Software . . . . . 14

**3 Grundlagen der ARM Architektur** 17

3.1 Begrifflichkeiten rund um den ARM Prozessor . . . . . 19

3.1.1 RISC und CISC . . . . . 19

3.1.2 Pipeline Prinzip . . . . . 20

3.1.3 Zwischenspeicherung in einem Cache . . . . . 23

3.1.4 MMU und MPU . . . . . 24

3.2 Kriterien zur Prozessorwahl . . . . . 26

3.3 AT91RM9200 Mikrocontroller . . . . . 29

3.3.1 Bootreihenfolge des AT91RM9200 . . . . . 31

3.3.2 ARM920T Mikroprozessor . . . . . 34

3.3.3 ARM9TDMI Prozessorkern . . . . . 37

**4 Layout eines eigenen Entwicklungsboards 41**
4.1 Spannungsversorgung des Entwicklungsboards . . . . . 42
4.1.1 Funktionsweise eines Step Down Wandlers . . . . . 42
4.1.2 Implementierung des Step Down Wandlers . . . . . 45
4.2 Beschaltung des AT91RM9200 Mikrocontrollers . . . . . 52
4.2.1 Beschaltung mit Quarzen . . . . . 53
4.2.2 Beschaltung der Phase Locked Loop (PLL) Baugruppen . . . 54
4.2.3 Implementierung der Reset Schaltung . . . . . 56
4.2.4 Implementierung des Boot Mode Select (BMS) Schalters . . . 57
4.3 Ethernet Physical-Layer Baustein . . . . . 58
4.4 Anschluss der parallelen Peripherie . . . . . 60
4.4.1 Anschluss eines „synchron DRAM (SDRAM)“ Bausteins . . . 60
4.4.2 Anschluss des NOR-Flashs . . . . . 61
4.4.3 Anschluss des Graphik Bausteins . . . . . 62
4.4.4 Erweiterung mit parallelen Peripheriebausteinen . . . . . 63
4.5 Anschluss von seriellen Peripheriebausteinen . . . . . 64
4.5.1 Anschlussmöglichkeiten der Universal Asynchronous Receiver Transmitter (UART) Schnittstellen . . . . . 64
4.5.2 Anschlussmöglichkeiten des Inter-Integrated Circuit (I2C) Busses . . . . . 66
4.5.3 Anschlussmöglichkeiten des Serial Peripheral Interface (SPI) . 66

**5 Grundlagen der verwendeten Software 69**
5.1 GNU binutils . . . . . 69
5.2 Die GNU Compiler Collection (GCC) . . . . . 71
5.3 Der GNU Debugger (GDB) . . . . . 73
5.4 Die C-Bibliothek . . . . . 75
5.4.1 Die GNU libc . . . . . 76
5.4.2 Die uClibc . . . . . 76
5.5 Abläufe automatisieren mit „make“ . . . . . 77
5.6 „Das U-Boot“ - Der Bootloader . . . . . 79
5.6.1 Dateistruktur des Quellcodes . . . . . 80
5.6.2 Konfigurieren des Bootloaders . . . . . 81
5.7 Der Linux Kernel 2.6 . . . . . 82
5.7.1 Komponenten des Kernels . . . . . 83

5.7.2 Dateistruktur des Quellcodes . . . . . 84
5.7.3 Erweitern des Quellcodes mit speziellen Patchsets . . . . . 85
5.7.4 Konfigurieren des Kernels . . . . . 86

**6 Konfiguration des Hostsystems 89**
6.1 Erzeugen einer Toolchain . . . . . 89
6.1.1 Kegel's crosstools . . . . . 91
6.1.2 uClibc buildroot . . . . . 93
6.1.3 Anpassungen an dem uClibc buildroot System . . . . . 95
6.2 Serielles Terminal . . . . . 96
6.2.1 Serielle Protokolle . . . . . 97
6.2.2 Programme zur seriellen Datenübertragung . . . . . 99
6.3 Der Trivial File Transfer Protocol (TFTP) Server . . . . . 102

**7 Erstellung und Funktionsweise der Entwicklungssoftware 105**
7.1 Der Bootprozess . . . . . 105
7.1.1 Erste Inbetriebnahme des Entwicklungsboardes . . . . . 105
7.1.2 Booten vom parallelen NOR Flash . . . . . 111
7.2 Durchgeführte Anpassungen für „das U-Boot“ . . . . . 116
7.2.1 Änderungen des Make Systems . . . . . 116
7.2.2 Anpassungen von vorhandenem Quellcode . . . . . 117
7.2.3 Neu erstellte Dateien . . . . . 119
7.3 Durchgeführte Anpassungen des Linux Kernels . . . . . 120
7.4 Das Dateisystem . . . . . 124
7.5 Hinzufügen neuer Softwarepakete . . . . . 125

**8 Zusammenfassung 127**
8.1 Ergebnis . . . . . 127
8.2 Ausblick . . . . . 128

**A Steckerbelegungen des Entwicklungsboards I**

**B Kurze Vorführung einer Testsitzung mit dem GNU Debugger (GDB) VII**

**Literaturverzeichnis XIII**