Gliederung
- Titel
- µCLinux
- Geschichte
- Vorteile
- Unterschiede
- Architekturen
- Ziele
- Ablauf
- Stand
- Portierung
- Test
- Ausblick
- Quellen

Debugging mit ucLinux auf ARM-Prozessoren

- Thema der Diplomarbeit
- betreut durch Steffen Köhler

µCLinux

- µ für „micro“ und C für „Controller“
- sprich: „you-see-linux“
- Linuxportierung für Mikrocontroller ohne Memory Management Unit
- Open-Source-Projekt
- basiert auf Linux-Kernel

Geschichte

- erste Portierung auf Motorola MC68328 (1998/99)
- erstes Zielsystem: PalmPilot
- entstand aus Linux-2.0 Kernel
- heute: komplette Distribution, enthält Linux-2.6, Anwendungen, Bibliotheken und Tool-Chains

Vorteile

- komplette Funktionalität des aktuellen Linux-Kernels (Treiber, Dateisysteme, ROMFS, Konsole)
- geringere Imagegröße (< 300 KB)
- optimierte C und C++ Bibliotheken
- uneingeschränkter Systemzugriff für Anwendungen (auch Register)

Unterschiede zu Linux

- betreffen Speicherverwaltung
- kein fork(), sondern vfork() → Multitasking
- kein wachsender Stack → mmap()
- kein Speicherschutz (protected memory)
- andere Bibliotheken

Unterstützte Architekturen

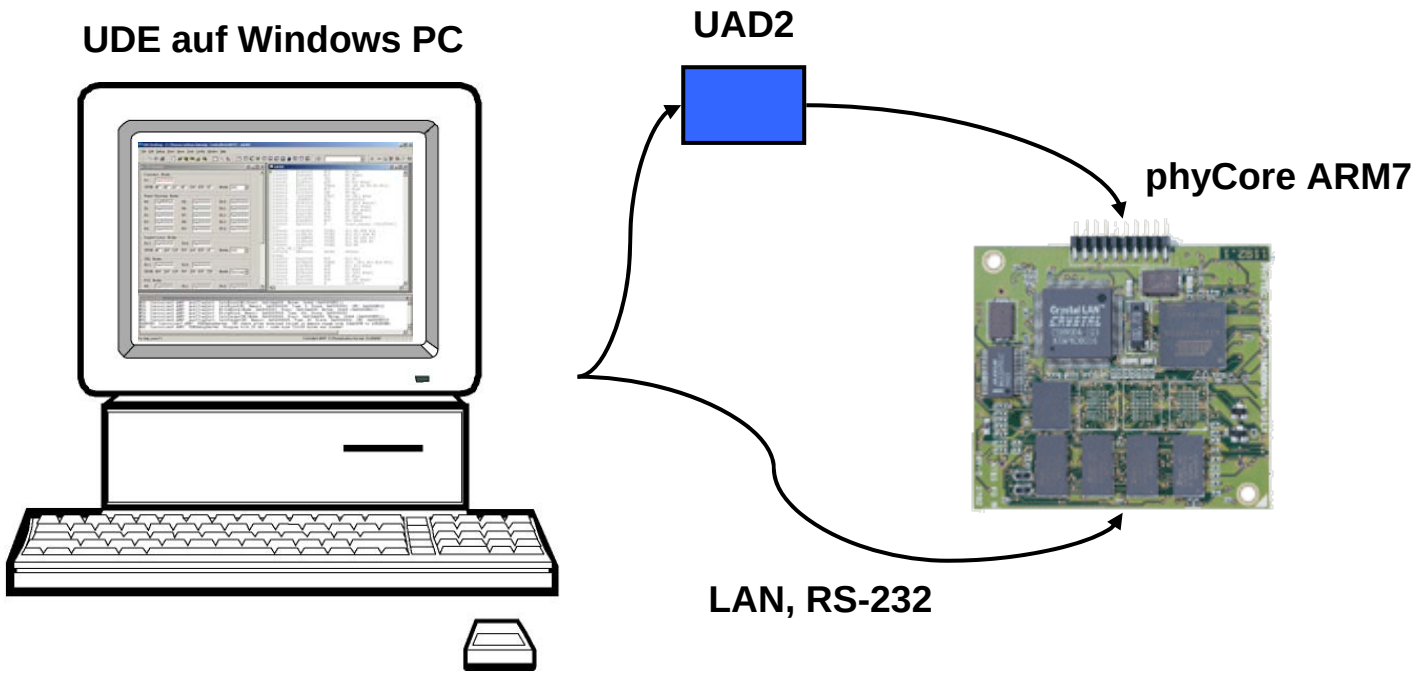
- Motorola Dragonfire, Coldfire
- ARM7TDMI, (ARM9)
- ATARI
- Xilinx Microblaze
- T-Concept XI524 DSL
- T-Eumex 630 LAN/DSL
- µCsimm

Ziel der Diplomarbeit

- lauffähiges µCLinux auf Entwicklerboard installieren
- Verbindung mit UDE (Windows) herstellen (UAD2, RS232, LAN)
- Anwendungen unter µCLinux ausführen und debuggen
- Kontrolle und Ausgabe über UDE

Gliederung
- Titel
- µCLinux
- Geschichte
- Vorteile
- Unterschiede
- Architekturen
- Ziele
- Ablauf
- Stand
- Portierung
- Test
- Ausblick
- Quellen

Ablauf



Bisheriger Stand

- Entwicklerboard phyCore AT91M55800A von Phytec
- angepasster µCLinux-Kernel auf Basis der Sourcen für Atmel AT91
- Download des Kernel als Image (.elf) auf Board über UDE und UAD/UAD2 funktioniert

Portierung und Anpassung

- Portierung unter Debian / Cygwin
- Installation einer aktuellen µCLinux-Distribution (15.12.2004)
- Installation einer Toolchain für ARM/.elf auf Basis von gcc
- Download aktueller Kernelsources von µCLinux/ARM2.6-Projekt

Gliederung
- Titel
- µCLinux
- Geschichte
- Vorteile
- Unterschiede
- Architekturen
- Ziele
- Ablauf
- Stand
- Portierung
- Test
- Ausblick
- Quellen

Portierung und Anpassung

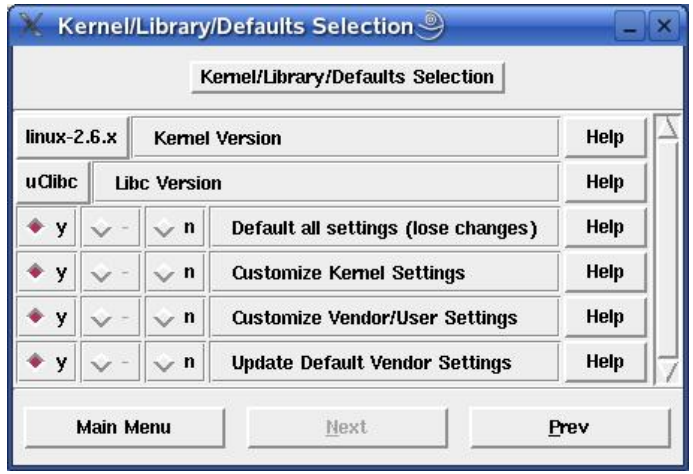
- Erstellen eigener Plattform auf Basis von Atmel‘s AT91
- Anpassen des Quellcodes
  - RS-232 Schnittstelle für Terminal
  - RAM- und Flashgrößen, Takt
  - eigener Interrupthandler
  - Treiber für Netzwerkschnittstelle

Portierung und Anpassung

- Quellen der „userland“-Umgebung anpassen
- Kernel konfigurieren und erstellen
- „userland“-Anwendungen erstellen
- Images in .elf-Format umwandeln und verknüpfen

Gliederung
- Titel
- µCLinux
- Geschichte
- Vorteile
- Unterschiede
- Architekturen
- Ziele
- Ablauf
- Stand
- Portierung
- Test
- Ausblick
- Quellen

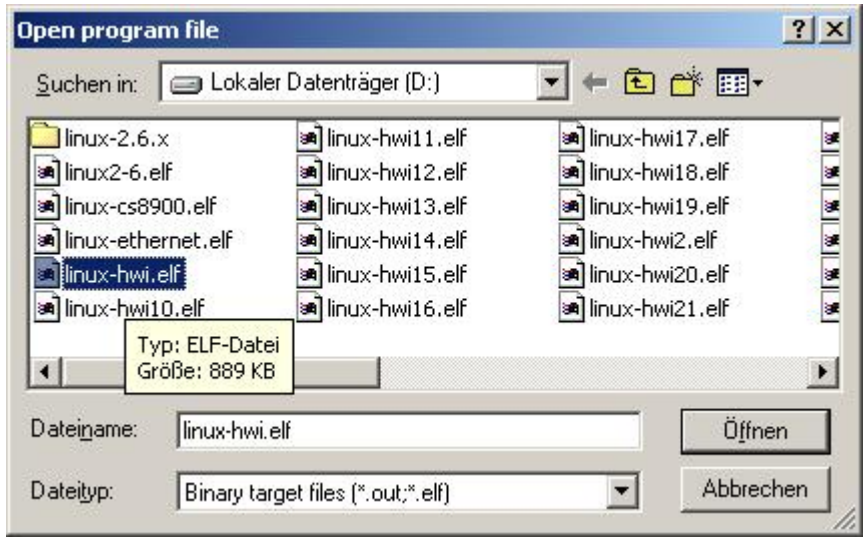
Portierung und Anpassung



Gliederung
- Titel
- µCLinux
- Geschichte
- Vorteile
- Unterschiede
- Architekturen
- Ziele
- Ablauf
- Stand
- Portierung
- Test
- Ausblick
- Quellen

Testen

- Download des erstellten Images auf das Versuchsboard per UDE

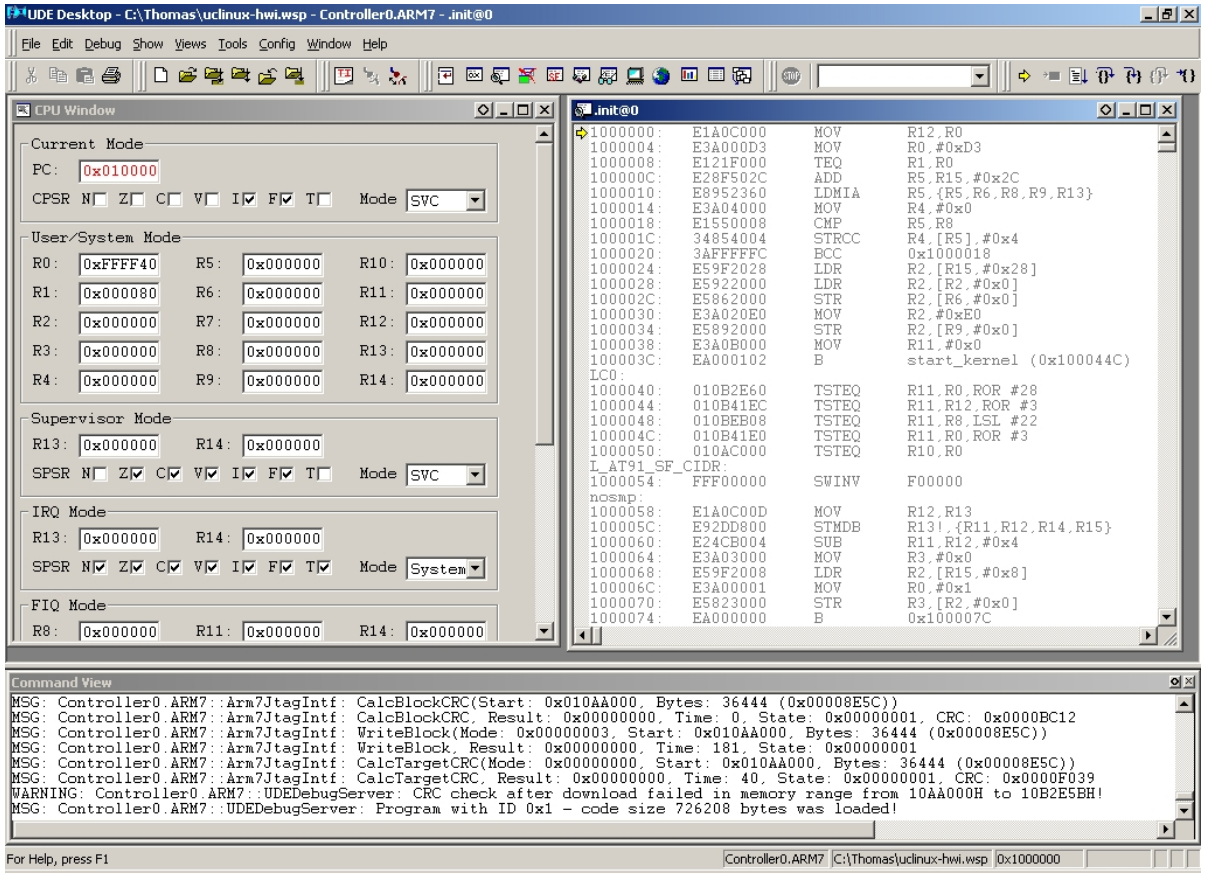


Testen

- Prozessor starten
- Kernel-Debugging per UDE
- bei Fehler Registerinhalte und Befehlszähler notieren, Speicherinhalte überprüfen
- Terminal gibt Kernelmeldungen aus
- Fehler beseitigen und neuen Kernel erstellen

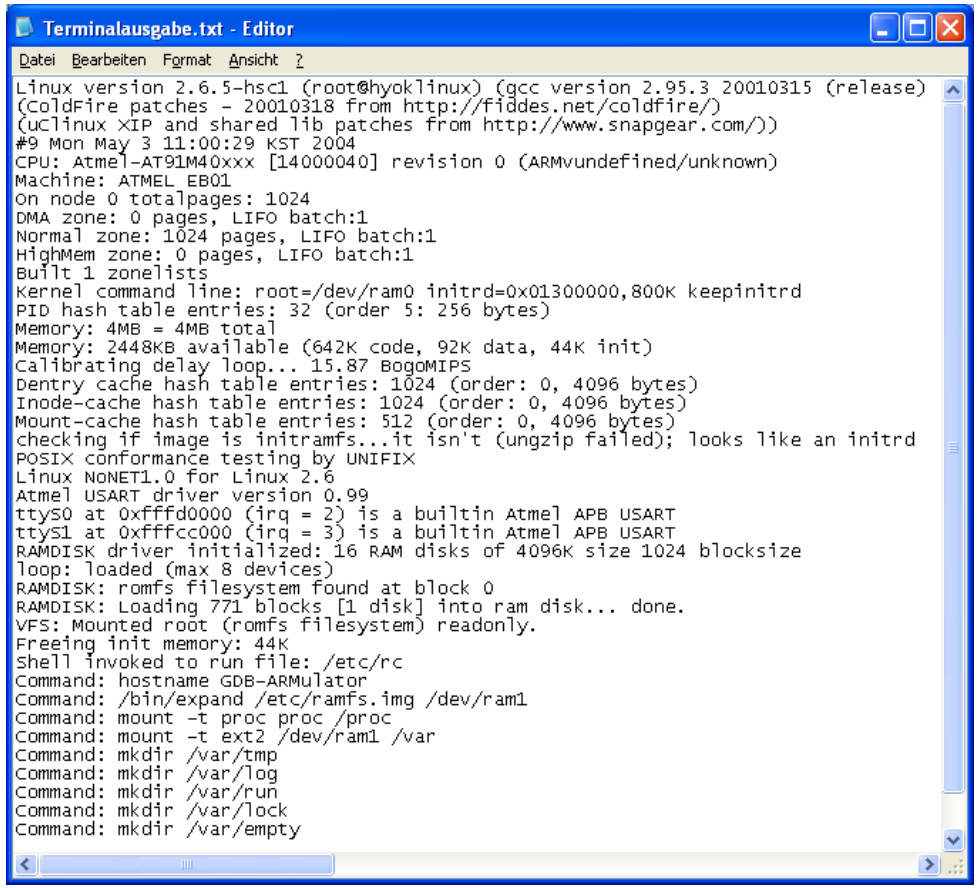
Gliederung
- Titel
- µCLinux
- Geschichte
- Vorteile
- Unterschiede
- Architekturen
- Ziele
- Ablauf
- Stand
- Portierung
- Test
- Ausblick
- Quellen

Testen



Gliederung
- Titel
- µCLinux
- Geschichte
- Vorteile
- Unterschiede
- Architekturen
- Ziele
- Ablauf
- Stand
- Portierung
- Test
- Ausblick
- Quellen

Testen



Ausblick

- Stabilität und Performance des µCLinux verbessern
- aktuellen GNU Compiler und Debugger für µCLinux erstellen und in „userland“-Image einbinden
- Netzwerktreiber verbessern
- Netzwerkkommunikation erweitern

Ausblick

- Terminal in UDE-Oberfläche integrieren
- Download einer µCLinux-Anwendung per Netzwerk
- Debuggermeldungen in UDE darstellen bzw. verwerten

Quellen

- http://www.uclinux.org
- http://www.ucdot.org
- µCLinux/ARM2.6-Projekt betreut von Hyok S. Choi http://opensrc.sec.samsung.com
- http://www.phytec.com