

Untersuchungen zum Aufbau eines echtzeitfähigen Embedded-Linux-Moduls

Gliederung

1. Einführung

Ziele

Begriffe

2. Linux und Echtzeit

3. Praktische Untersuchungen

Überblick

Beispiel: Timer

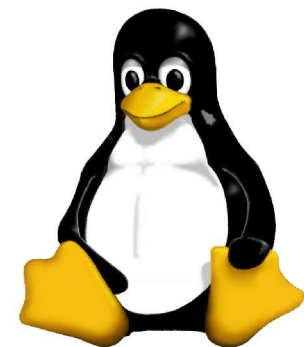
Messungen der Echtzeitperformance

4. Zusammenfassung

1. Einführung

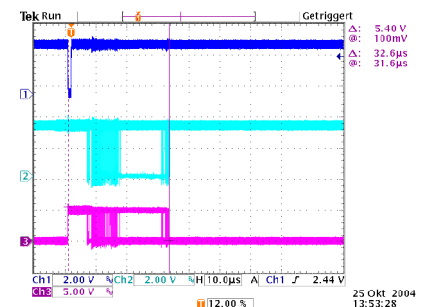
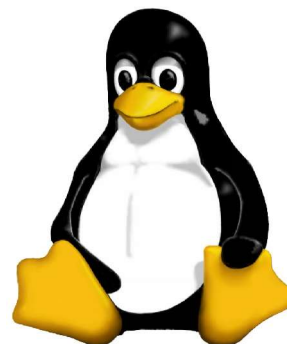
- Eingebettete Systeme weit verbreitet
- Steuer- und Regelfunktionen
- Echtzeitaufgaben

- Zunehmend auch Linux
- Vorteil: Universelles OS
- Nachteil: Keine Echtzeit mit Standard-Linux

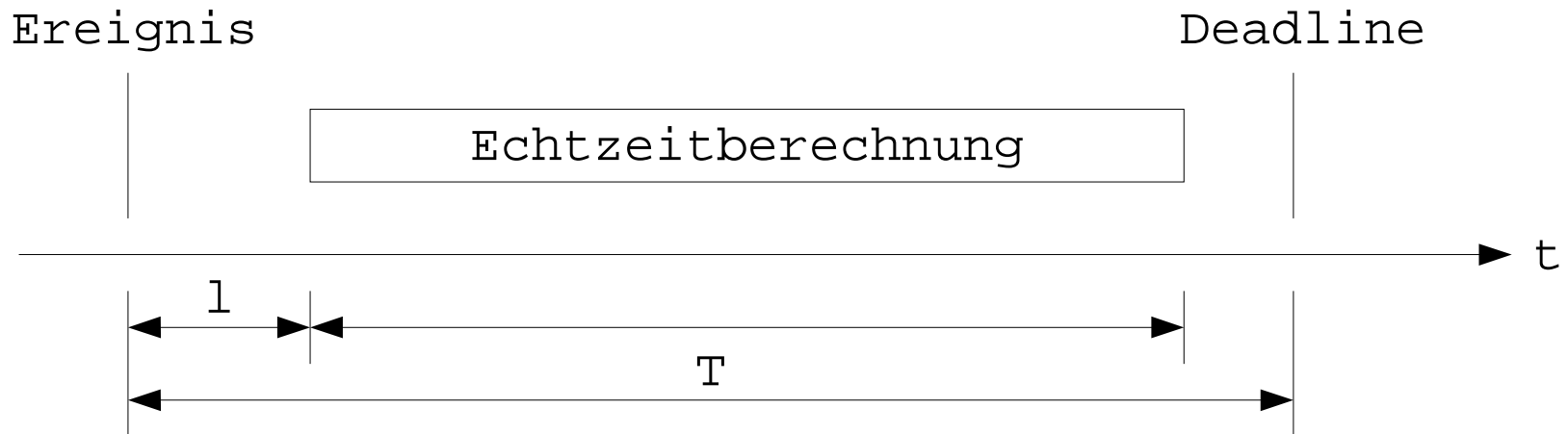


Ziele

- Möglichkeiten für RT und Linux finden
- Eignung für Embedded-Linux (kein x86)
- Praktische Umsetzung der Ergebnisse
- Performanceaussagen



Begriffe

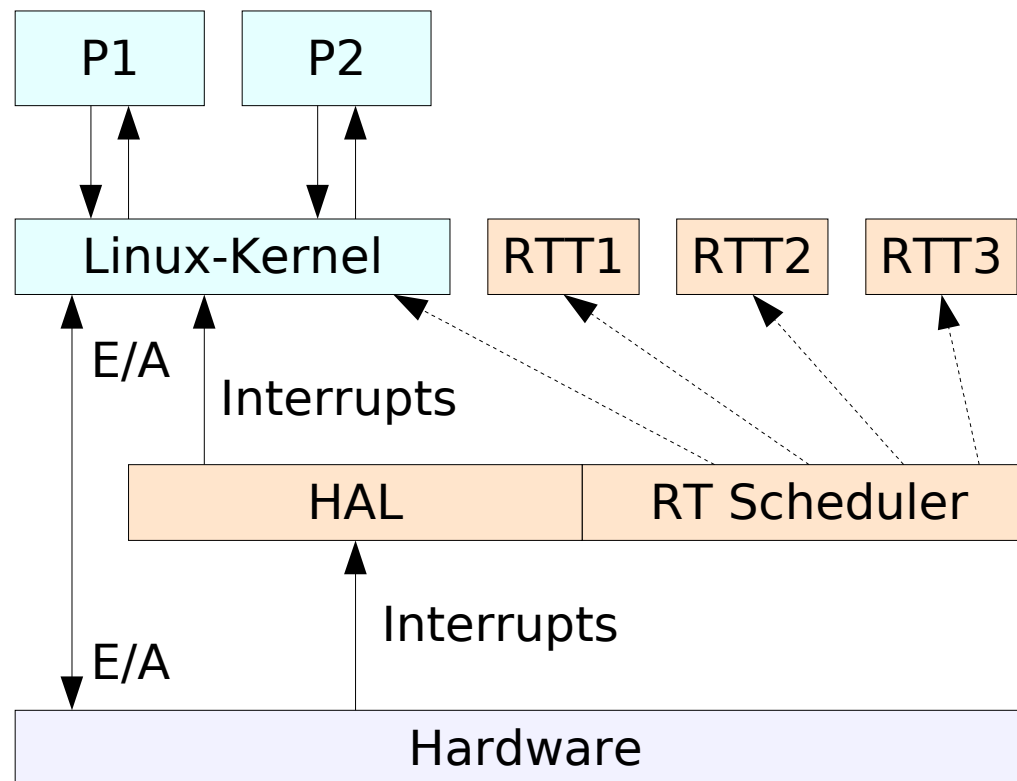


- Deadline
- Latenz l
- Jitter
- Echtzeit: deterministisches Zeitverhalten

2. Linux und Echtzeit

- Viele verschiedene Erweiterungen
- Freie und kommerzielle Lösungen
- Bewertung nach verschiedenen Kriterien
 - Machbarkeit
 - Nutzen
- Auswahl einer geeigneten Variante

3. Praktische Untersuchungen



- Mikrokernelarchitektur: HAL+Scheduler

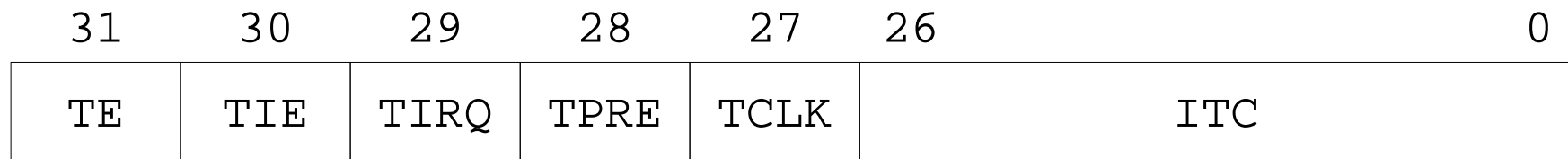
Überblick

- Hardware
- HAL und Scheduler
 - ➔ Linux-Kernel und RTAI-Module
 - ➔ Interruptverarbeitung
 - ➔ Taskwechsel
 - ➔ Timerfunktionen
- IPC
- Testsuite

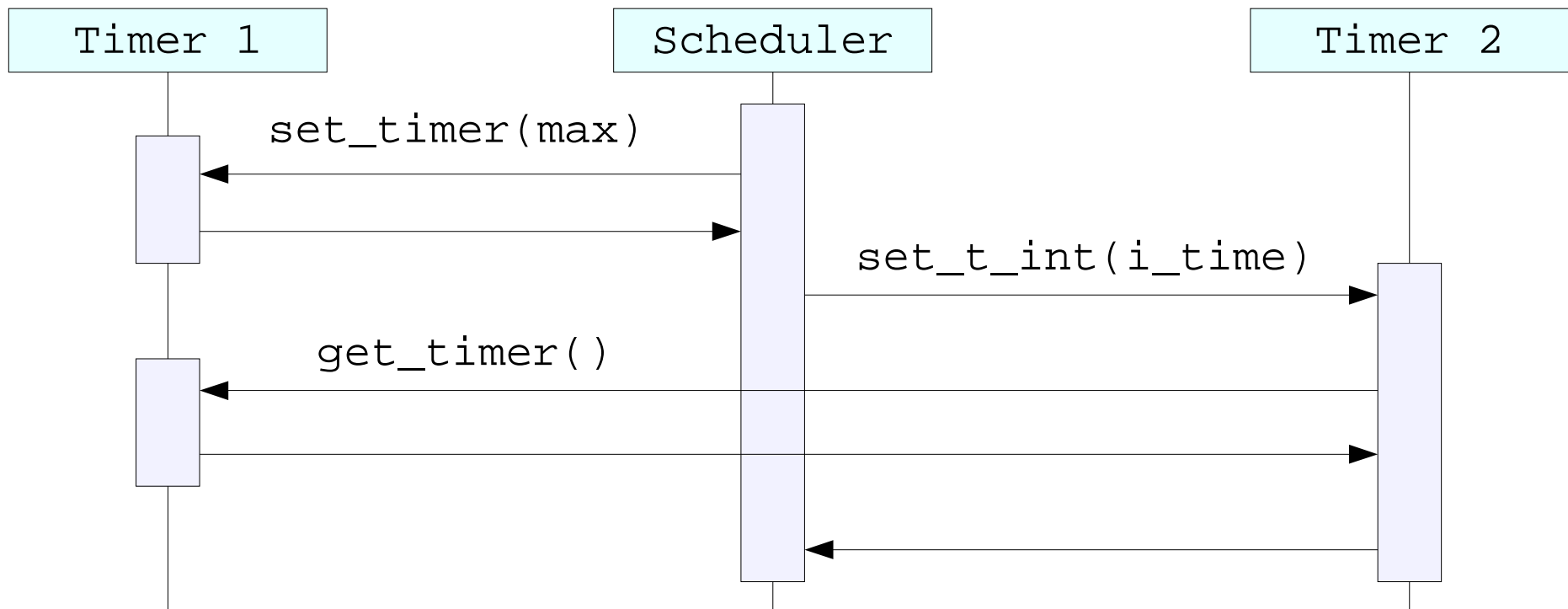
CPU	NET+50
CPU-Takt	44 MHz
Speicher	16 MB SD
Flash	4 MB
Netz	10/100
OS	uClinux

Beispiel: ns-Timer

- Grundbestandteil des RTAI-Schedulers
- X86: TSC-Register, APIC oder 8254
- ARM-CPU: verschiedene Timer-Implementierungen
- NET+50: zwei 27 Bit Counter

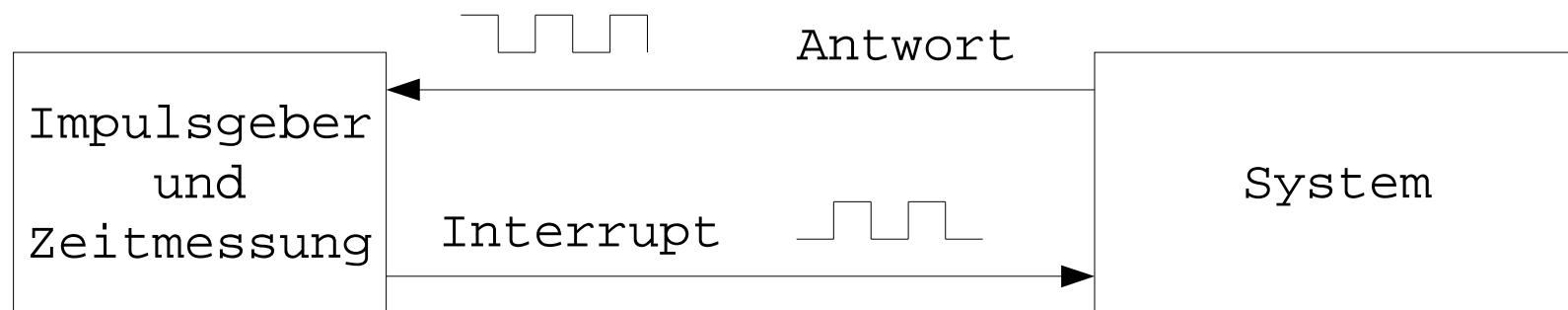


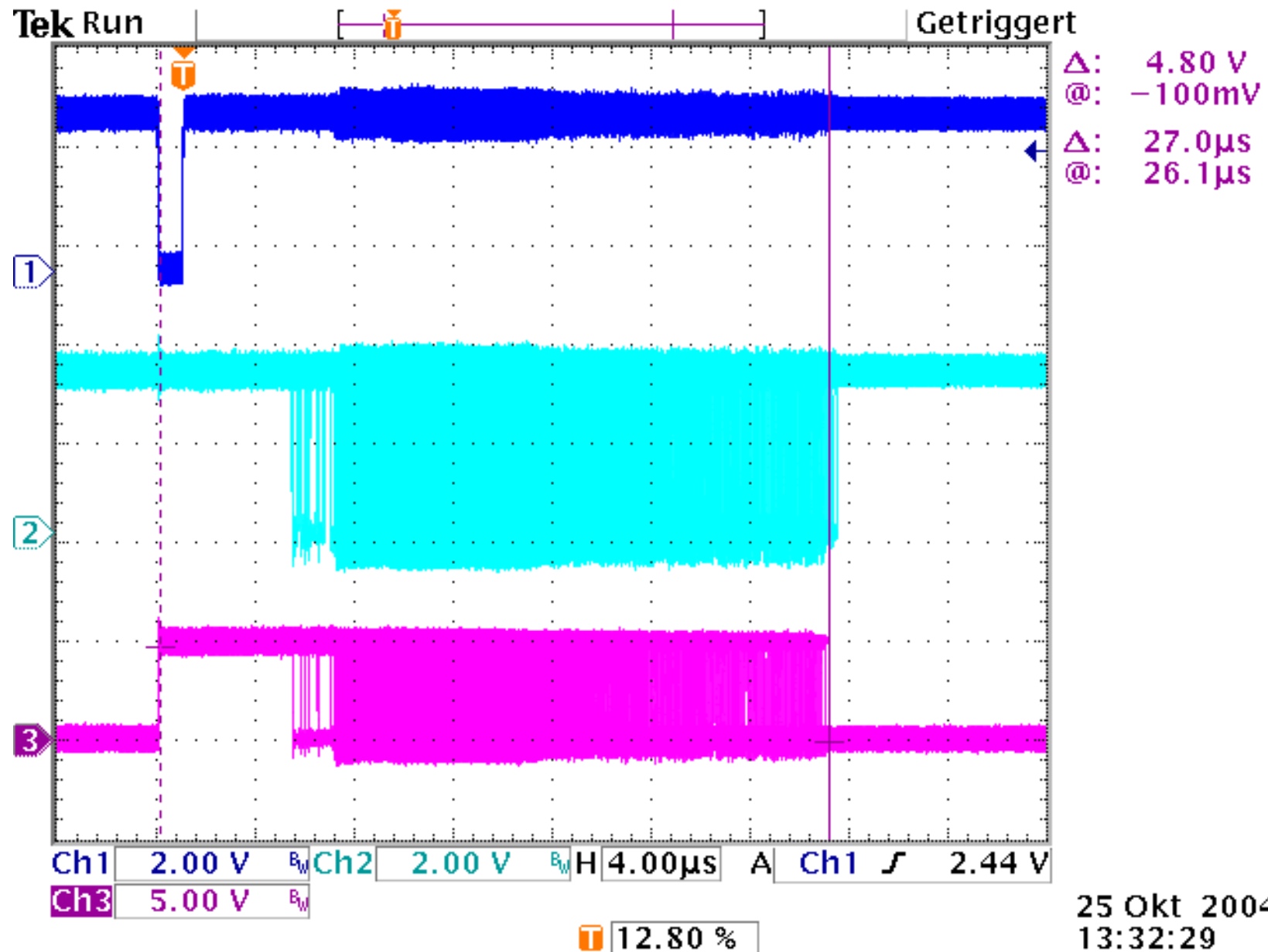
- Verwendung eines einzelnen Timers führt zu Problemen
- z.B. Ungenauigkeiten beim erneuten Setzen
- Lösung: beide Timer benutzen

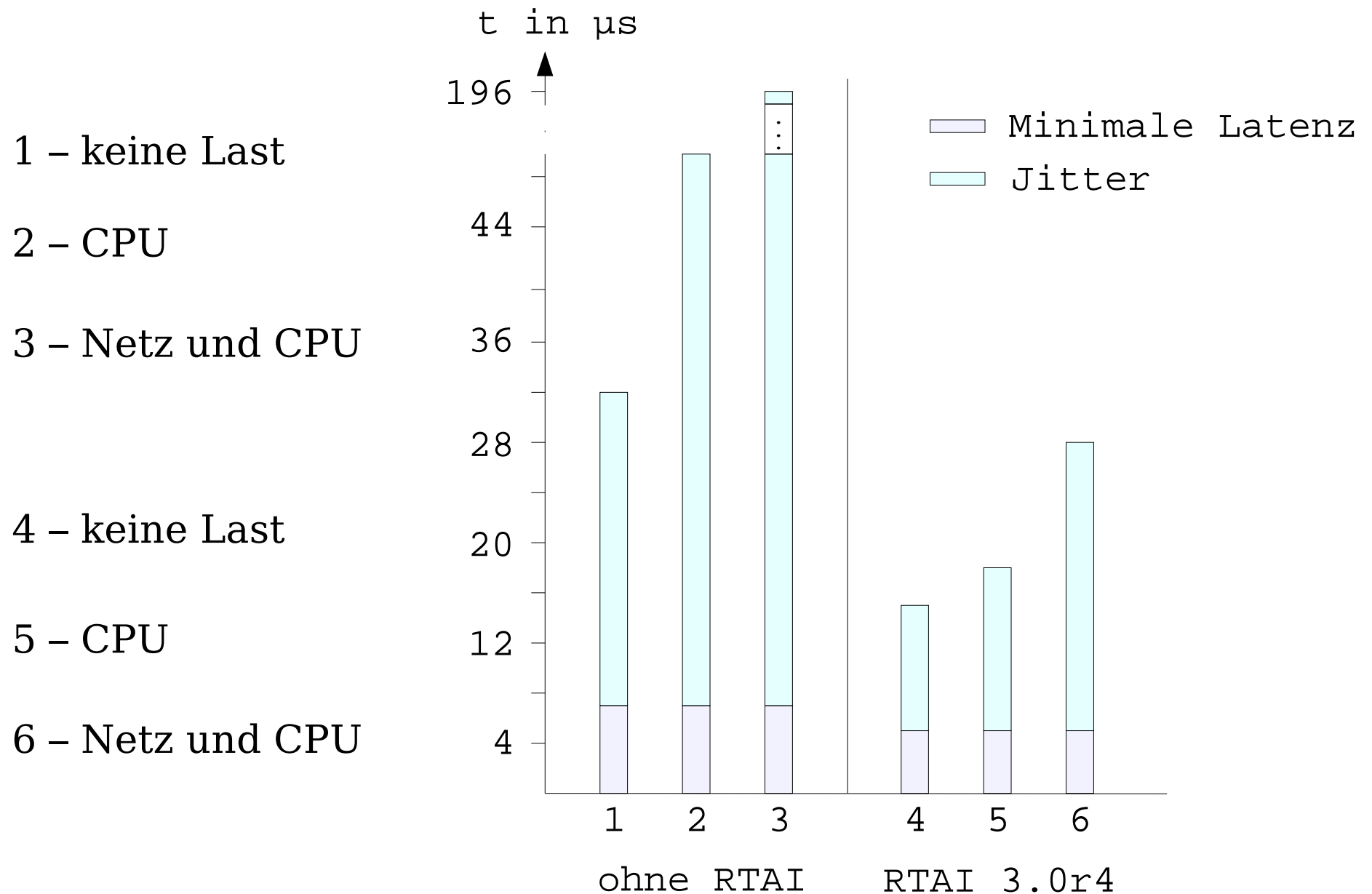


Messungen

- Statistische Ergebnisse
- Vergleich mit x86
- RTAI-Testsuite: Scheduler Latenzen und Jitter, Zeit für Taskwechsel
- Messungen mit externer Hardware







4. Zusammenfassung und Ausblick

- Embedded-Linux und Echtzeit mittels Erweiterung
- RTAI bietet gute Voraussetzungen
- Umsetzung auf ARM7-Modul
- Gute Echtzeitperformance auch auf eingebetteten Systemen
- Einsatz für Regel- und Steuerungsaufgaben, Signalerfassung
- Echtzeitvernetzung: verteilte Systeme

Kontakt:

Dipl.-Inf. Thomas Elste

Institut fuer Mikroelektronik- und Mechatronik-Systeme gGmbH
Ehrenbergstraße 27, 98693 Ilmenau, Germany

Email: thomas.elste@imms.de WWW: <http://www.imms.de>
Tel: +49 (0) 3677 695543 Fax: +49 (0) 3677 695515