

Neue DSP-Hard- und Softwarelösungen für den Einsatz in Mehrkoordinaten- Nanomess- und Positioniersystemen

**Christian Uhle, Bernd Däne,
Todor Vangelov, Wolfgang Fengler**

TU Ilmenau, Fachgebiet Rechnerarchitekturen

Gliederung

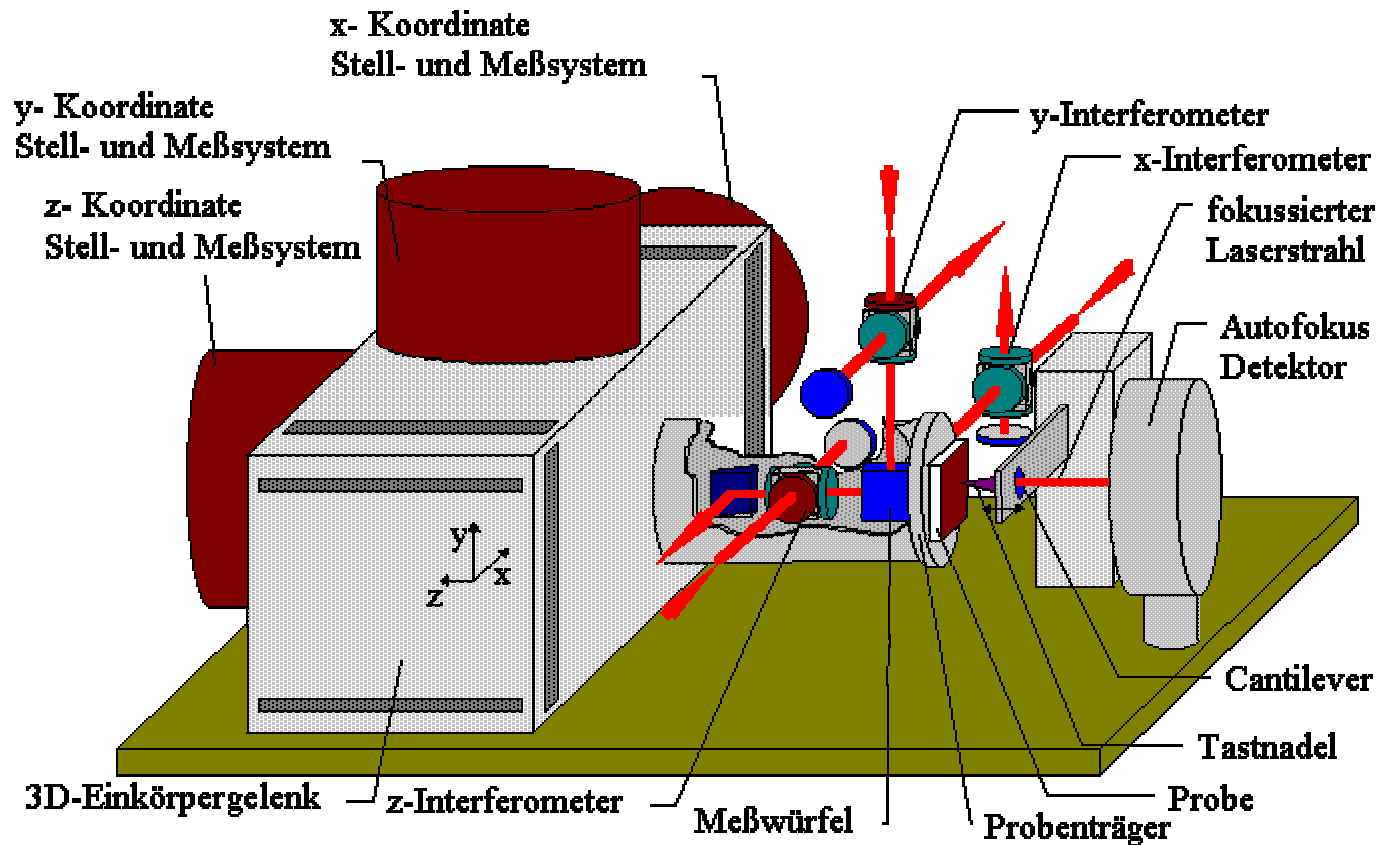
1. Einleitung
2. Gestaltung der Hardware
3. Gestaltung der Software
4. Ergebnisse
5. Ausblick

1. Einleitung

- Nanomess- und Positioniersysteme:
 - Einrichtungen zur schnellen Positionierung und Positionsbestimmung mit Auflösungen im Nanometerbereich
- Anwendungsbeispiele:
 - Bearbeitungs- und Messaufgaben in der Mikrosystemtechnik
 - Atomkraftmikroskope

Aufbau eines Atomkraftmikroskopes:

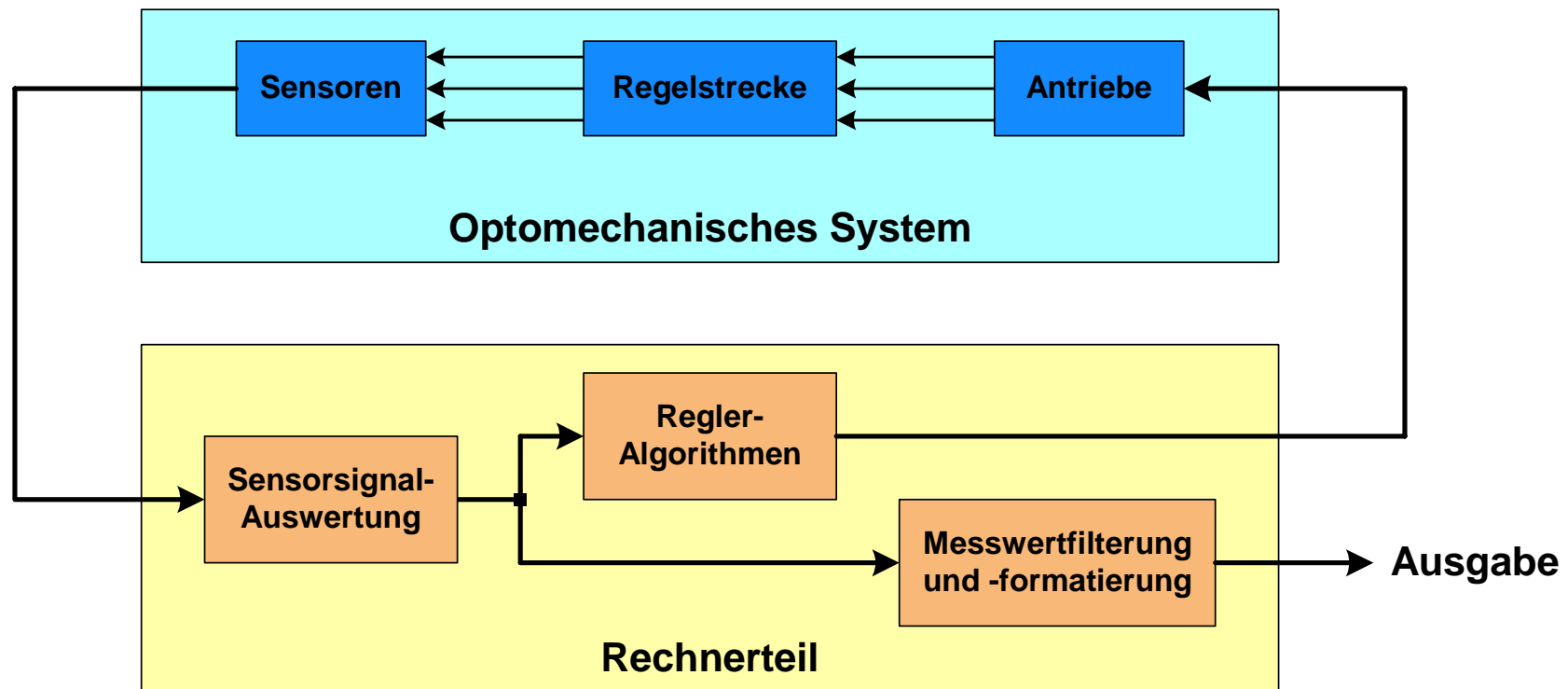
Demonstrator aus dem Fachgebiet Prozessmesstechnik der TU Ilmenau



Funktionsweise

- Interferenzoptische Positionsmessung in drei Achsen
- Lageregelung in geschlossenen Regelkreisen
- Sehr schnelle Messwertfilterung und Reglerberechnung durch eingebettetes Rechnersystem

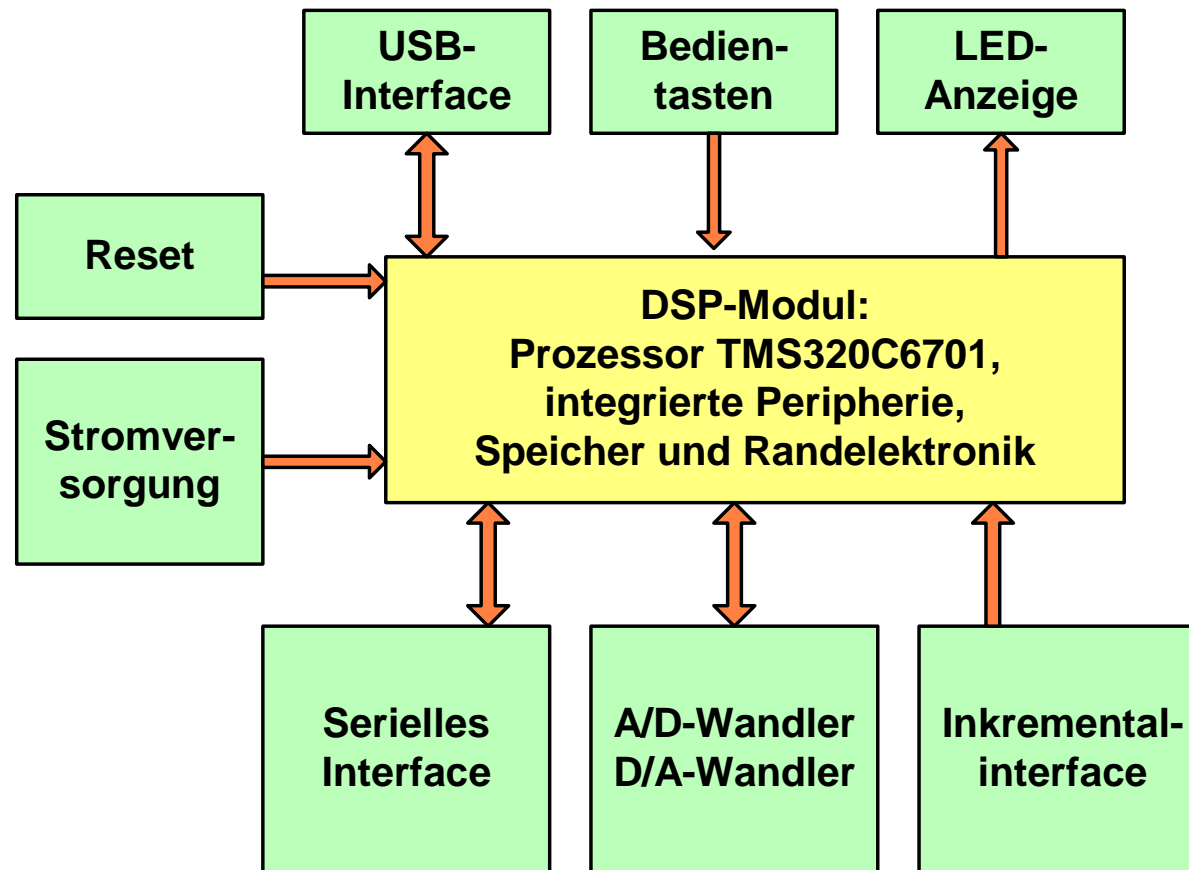
Signalfluss im Gesamtsystem



2. Gestaltung der Hardware

- Sehr schneller RISC-Signalprozessor mit VLIW-Architektur: TMS320-C6701 (Texas Instruments)
- Verwendung eines industriellen Kompaktmoduls mit Prozessor, Randelektronik und Speicher
- Experimentalaufbau zur Entwicklung von Funktionen wie Messwerterfassung und Hostkommunikation ohne Verfügbarkeit des Gesamtsystems

Blockschaltbild des Experimentalaufbaus



3. Gestaltung der Software

- Eigenes Echtzeitbetriebssystem „eRTOS“
- Applikationstasks für Digitalfilter, Host-Kommunikation, Ablauforganisation usw.
- Spezielle Optimierung zeitkritischer Teile
- Modulare Erweiterbarkeit und Modifizierbarkeit

Grundanforderungen an das Echtzeitbetriebsystem eRTOS

- Modularität
 - Beschränkung auf wirklich benötigte Funktionen (z.B. Tasksteuerung, Synchronisation, Nachrichtensystem)
 - Schlankes Applikationsinterface
 - Spezielle Unterstützung für Tasks, die in einem festen Zyklus laufen müssen
 - Geringer Ressourcenverbrauch unter Beachtung der Eigenheiten des verwendeten Prozessors
- ➡ Kompromiss von Funktion zu Ressourcenbedarf

Zwei Typen von Tasks

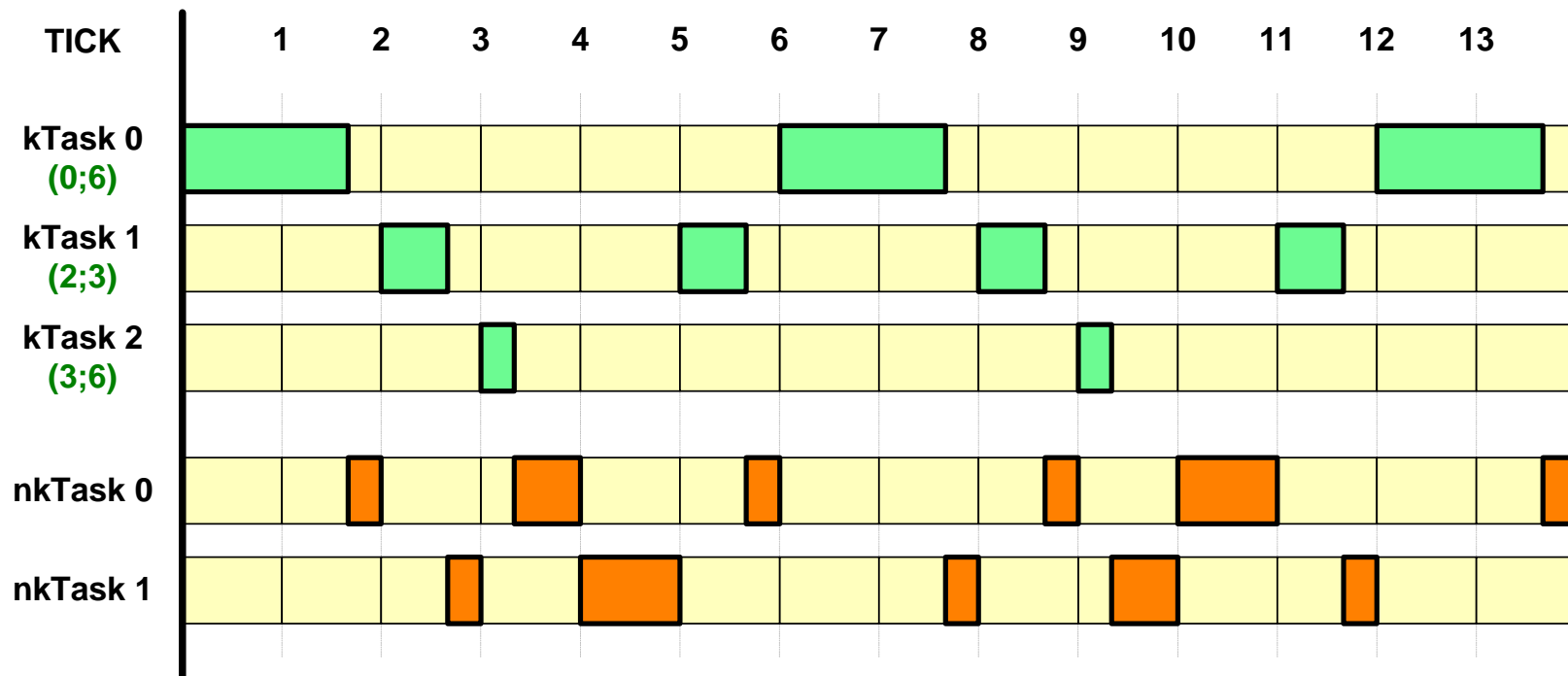
„kTask“ (kooperierend)

- Start in festen Intervallen
- keine Preemption
- konstanter Zeitbedarf
- bevorrechtigt
- z.B. für Filteralgorithmen

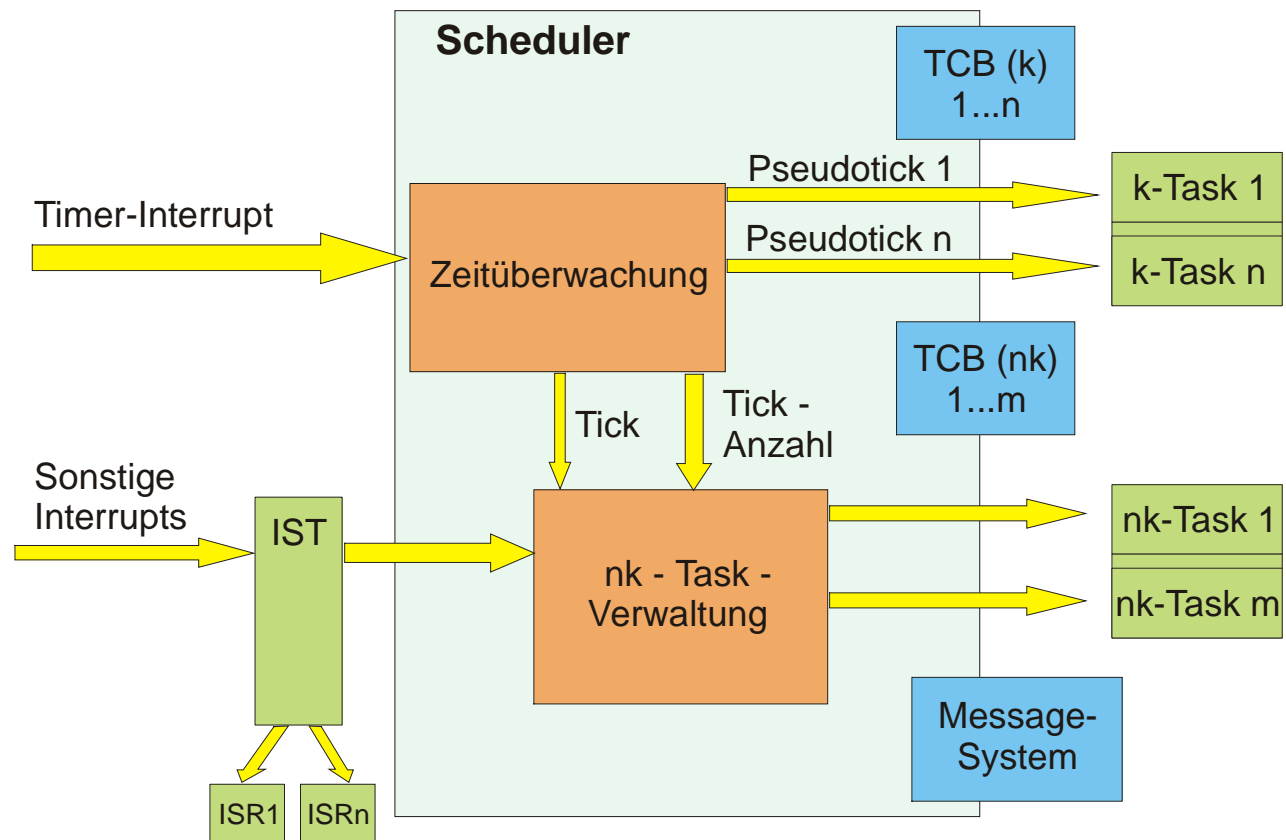
„nkTask“ (nicht kooperierend)

- preemptives Scheduling
- Start bei:
 - Tick
 - Ende einer kTask
 - Suspendierung einer nkTask
- z.B. für Bedien- und Kommunikationsvorgänge

Beispiel für das Scheduling der Tasks



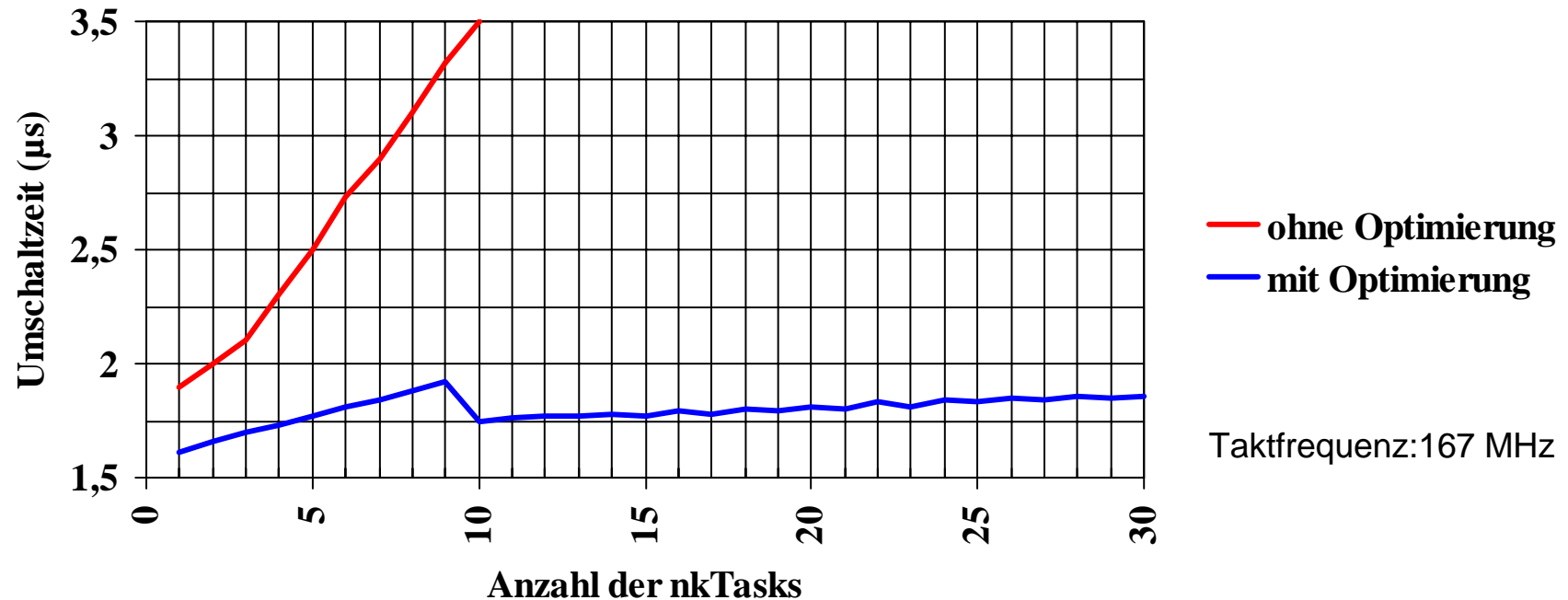
Struktur des Echtzeitbetriebssystems eRTOS



4. Ergebnisse

- Aktueller Entwicklungsstand:
 - Erste Version fertiggestellt
 - Untersuchungen zu Performance und Stabilität
 - Vorbereitung der nächsten Versionen
- Kernpunkte:
 - Einfluss des Betriebssystems auf die Performance der Filter- und Reglerfunktionen
 - Testweise Einbindung unterschiedlicher E/A-Funktionen und sonstiger Zusatzfunktionen

Umschaltzeiten in Abhängigkeit von der Anzahl der Tasks (gemessen)



Spezielle Erkenntnisse

- Flexibilität schluckt Ressourcen.
- Tick-Länge nicht kürzer als nötig einstellen.
- E/A-Funktionen kosten Zeit.
Verbesserung durch:
 - Blockbildung
 - DMA-Benutzung
- Speicher-Mapping muss geplant werden.
- Handoptimierung ist stellenweise unverzichtbar.

5. Ausblick

- Weiterführung als Kooperationsthema (gefördert vom Land Thüringen)
- Stärkere Einbeziehung formaler Entwurfs- und Testverfahren
- Weiterentwicklung bezüglich Messgenauigkeit und Dynamik
- Nutzung von Ergebnissen auch für verwandte Projekte