45th International Scientific Colloquium
Ilmenau Technical University
October 04 - 06, 2000

Christian Uhle / Bernd Däne / Todor Vangelov / Wolfgang Fengler

Neue DSP-Hardware- und Softwarelösungen für den Einsatz in Mehrkoordinaten-Nanomeß- und Positioniersystemen

Abstrakt

Im Beitrag wird die Entwicklung einer Verarbeitungs- und Steuereinheit für Mehrkoordinaten-Nanomeß- und Positioniersysteme beschrieben. Dabei handelt es sich um ein Stand-alone-System aus Modulen mit einem digitalen Signalprozessor TMS320C6701 von Texas Instruments in Piggy-Back-Technologie, dessen Bedienung über einen Host-PC geschieht. Für einen frei konfigurierbaren und modular erweiterbaren Einsatz des Systems baut die Software auf einem neu entwickelten echtzeitfähigen Multitasking-Betriebssystem auf.

1. Einführung

Gegenwärtig steigen die Anforderungen an die Meßtechnik für die Mehrkoordinatenmessung und Positioniertechnik im Submikrometer- und Nanometerbereich. Durch die fortschreitende Tendenz zur Miniaturisierung und die damit verbundene Erhöhung der Präzisionsanforderungen in der Mikrosystemtechnik und in der Mikromechanik wird eine qualitative Weiterentwicklung der Präzisionsmeß- und Positioniertechnik erforderlich [1]. Im Rahmen des Verbundprojektes "Mehrkoordinaten-Nanomeß- und Positioniersystem" wird eine neue Meßgerätegeneration für Messungen im Submikrometer- und Nanometerbereich entwickelt. Meßtechnische Grundlage dieser Geräte sind lichtwellenleitergekoppelte interferenzoptische Sensoren. Gegenstand dieses Beitrags ist eine Hard- und Softwarelösung für die Meßwertverarbeitung und Steuerung dieser Geräte.

2. Anforderungen an die Verarbeitungs- und Steuereinheit

2.1. Anforderungen der Meßwertverarbeitung

Die Verarbeitungs- und Steuereinheit hat Aufgaben der Bedienung und Steuerung des Meß- und Positioniersystems zu erfüllen, die Meßwerte zu erfassen und Algorithmen auf diese Meßwerte anzuwenden. Dabei werden große Datenmengen verarbeitet: Zur Fehlerkorrektur soll unter anderem eine digitale Filterung in einer Größenordnung von 100 Filterkoeffizienten durchgeführt werden. Bei der Verarbeitung der Meßwerte wird mit einer hohen Auflösung (32-bit-integer-Meßwerte) und einer hohen Abtastrate gearbeitet (100 kHz).

2.2. Anforderungen an die Konfiguration

Die entwickelte Verarbeitungs- und Steuereinheit wird für eine Meßgerätefamilie entwickelt, zu der verschiedene Meßgeräte gehören werden. Dazu gehören Positioniersysteme, Tastschnittgeräte und Distanzmeßgeräte. Deshalb soll die Funktion der Verarbeitungs- und Steuereinheit je nach Anforderung frei konfigurierbar und modular erweiterbar sein.
Die softwaretechnische Grundlage für die anwendungsspezifische Konfiguration und die modulare Erweiterbarkeit bildet das entwickelte Echtzeitbetriebssystem eRTOS.

3. Beschreibung der Hardware

Als Hardware der Verarbeitungs- und Steuereinheit werden die in Piggy-back-Technologie ausgeführten D.Module C6701 von D.SignT eingesetzt. Dabei handelt es sich um eine vorkonfektionierte Lieferform von Prozessormodulen mit Standardkomponenten (zum Beispiel Speicher) und Verbindungstechnik. Diese Technologie hilft, Entwurfskosten zu senken und führt dazu, daß beispielsweise SMD-Handhabungstechnik auf Anwenderseite eingespart werden kann. Die hardwareseitige Erweiterung geschieht durch das Aufstecken passender Module auf oder unter das Prozessormodul. Die als Rechnerhardware gewählten Stand-alone-Module sind mit jeweils einem DSP TMS320C6701 von Texas Instruments und integrierter Peripherie zur Meßwerterfassung ausgerüstet. Die Module sind im Scheckkartenformat in industrieller SMD-Technologie gefertigt.

3.1. Eigenschaften des verwendeten Prozessor-Typs

Als Prozessor für die Module wird ein DSP TMS320C6701 von Texas Instruments eingesetzt. Dabei handelt es sich um einen mit maximal 200 MHz getakteten Signalprozessor mit VelociTI-Architektur [2]. Die Architektur implementiert das VLIW-Prinzip (Very Long Instruction Word, [3]). Der Prozessor verfügt über 8 funktionelle Einheiten und kann somit bis zu 8 Befehle parallel ausführen. Die neue Architektur wird besonders im Zusammenspiel mit den von Texas Instruments angebotenen hochoptimierenden Code Generation Tools (C-Compiler, Assembler, Assembly Optimizer) ausgenutzt. Bei der Wahl des Prozessortyps wurde zugunsten dieses Prozessors entschieden, um dem hohen Anspruch an die Rechenleistung zu genügen. Die parallele Abarbeitung von Befehlen ermöglicht es, den Echtzeitanforderungen gerecht zu werden.
Der interne Speicher ist in Datenspeicher und Programmspeicher (jeweils 64 kBytes) aufgeteilt. Der Prozessor verfügt weiterhin über 2 Multichannel Buffered Serial Ports (MCBSP), 2 Timer, 4 DMA-Controller und ein 16-bit Host-Interface. Die DMA-Controller spielen eine wichtige Rolle für die Datenaquisition. Da die Zugriffe auf den externen Bus, von dem die zu erfassenden Daten (Meßwerte) gelesen werden, langsam gegenüber der Verarbeitungsgeschwindigkeit des Prozessors sind, dienen die DMA-Controller dazu, die Daten einzuholen, ohne gleichzeitig Prozessorzeit zu verbrauchen.

3.2. On-board Peripherie

Speichermäßig sind die Module mit 256 kByte SBSRAM mit 0 Wait States und 512 kBytes nicht-flüchtigem Flash Memory ausgerüstet. Frei konfigurierbar als Bit-I/O-Port oder zum Anschluß weiterer peripherer Geräte ist ein PLD (programmierbares Logik-Baustein) auf dem Modul untergebracht.

Weitere Peripheriebausteine sind ein externes Bus-Interface zum Anschluß von externen Peripheriebaugruppen und Speichererweiterungen, eine serielle Schnittstelle, Wait-State-Generator, Spannungsversorgung, Spannungsüberwachung und Watchdog.

4. Beschreibung der Software

Gegenstand der Entwicklung am Fachgebiet Rechnerarchitektur sind die hardwarenahen Softwarekomponenten. Dazu gehören ein echtzeitfähiges Multitasking-Betriebssystem, die Implementation einer USB-Schnittstelle sowie die Implementation von Treibern zur Meßwertaquisition und Kommunikation mit weiterer Hardware. Weitere Hard- und Softwarekomponenten werden vom Fachgebiet Prozeßmeßtechnik der TU Ilmenau und vom Zentrum für Bild- und Signalverarbeitung Ilmenau entwickelt.

4.1. Warum ein eigenes Betriebssystem

Kernstück der Software ist das am Fachgebiet Rechnerarchitektur entwickelte Echtzeit-Multitasking-Betriebssystem eRTOS. Das modular erweiterbare Echtzeitbetriebssystem eRTOS wurde speziell für die digitale Signalverarbeitung entwickelt. Es unterstützt als Besonderheit zwei Arten des Taskmanagements:

- in festem Zeitraster laufende, nicht unterbrechbare Tasks, die speziell für Meßwertaquisition und -verarbeitung in Echtzeit benötigt werden (hier kooperierende Tasks genannt),
- unterbrechbare Tasks, die z.B. Bedienfunktionen bearbeiten und dabei die verbleibenden Zeitfenster nutzen (hier nichtkooperierende Tasks genannt).

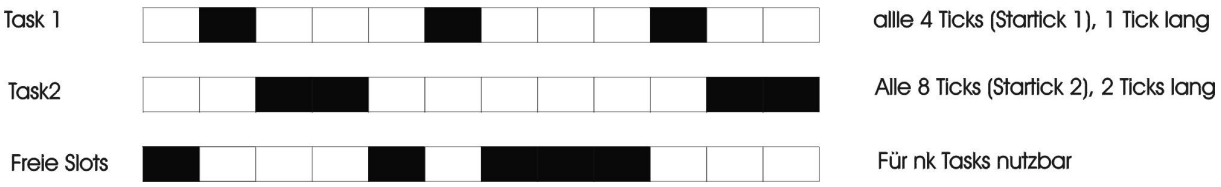

Abbildung 1: Scheduling der kooperierenden und nichtkooperierenden Tasks

Abb. 1 zeigt eine Beispielkonfiguration mit zwei kooperierenden Tasks (Task 1 und Task 2) und verbleibenden Slots für die nichtkooperierenden Tasks (unten).
Neben der oben beschriebenen Anforderung an eine anwendungsspezifische Konfiguration ermöglicht der modulare Einsatz des Echtzeitbetriebssystems dem Nutzer das Hinzufügen beziehungsweise Abwählen von Systemfunktionen wie zum Beispiel dem Nachrichtensystem. Dadurch können gegebenenfalls Ressourcen (an Speicher und Prozessorleistung) eingespart werden und eine leistungsstärkere Performance erreicht werden.
Nicht zuletzt konnten durch die Entwicklung eines eigenen Betriebssystems erhebliche Kosten eingespart werden, die durch den Kauf eines kommerziellen Betriebssystems einschließlich Entwicklungsumgebung entstanden wären.

4.2. Aufbau des Echtzeitbetriebssystems eRTOS

Grundsätzlich unterteilt sich die Struktur des Betriebssystems (siehe Abb. 2) in Systemkern (Taskverwaltung) und Applikationsinterface (Systemfunktionen).
Zur Taskverwaltung werden zwei Sorten Interrupts unterstützt: Timer-Interrupts und

Nutzer-Interrupts (sonstige Interrupts). Timer-Interrupts werden vom Betriebssystem selbst verwaltet (Scheduler) und sind unter anderem für das Starten der kooperierenden Tasks in einem festen Zeitraster verantwortlich. Die Nutzer-Interrupts können Systemfunktionen aufrufen oder nichtkooperiende Tasks starten. Der Scheduler schaltet in Abhängigkeit vom Timer-Tick die kooperierenden Tasks um und weist die freien Time-Slots den nichtkooperierenden Tasks zu.
Systemfunktionen ermöglichen das Abfragen der Taskeigenschaften (ID, Task-Zustand usw.), das Suspendieren und Entsperren einer Task und das Austauschen von Nachrichten zwischen den Tasks.

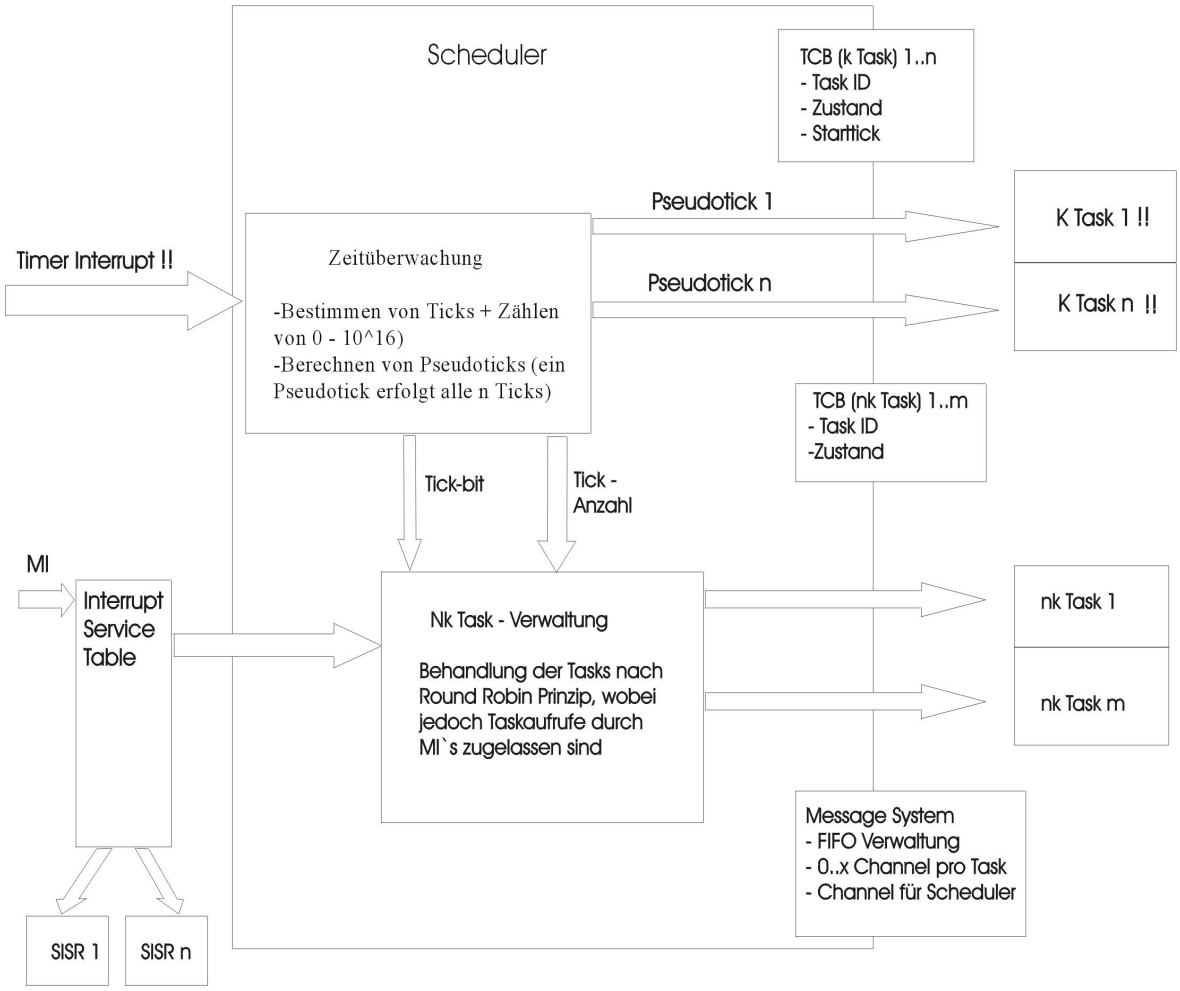

Abbildung 2: Struktur des Echtzeit-Multitasking-Betriebssystems eRTOS

4.3. Konfiguration und Einbinden der Nutzer-Tasks

Bei der Konfiguration des Betriebssystems legt der Programmierer fest, welche Tasks existieren, in welchem Zeitraster diese laufen sollen und welche der wählbaren Systemfunktionen des Betriebssystem genutzt werden. Weiterhin wird die Länge des Systemticks eingestellt und damit bestimmt, wieviel Zeit den kooperierenden Tasks für ihre Ausführung zur Verfügung steht. Deshalb ist ein vorheriges Profiling der kooperienden Tasks notwendig, um deren Zeitverbrauch zu messen.

4.4. Kenngrößen vom eRTOS

4.4.1. Das Zeitverhalten

Um den Tasks ein Maximum an Rechenzeit zur Verfügung zu stellen, wurden häufig benutzte Kernroutinen und Systemrufe per Hand auf Laufzeit optimiert. Die Effektivität eines Echtzeit-Multitasking-Betriebssystems wird vor allem durch zwei Zeiten charakterisiert: der Zeitverbrauch des Task-Kontextwechsels und die Reaktionszeit bei Nutzerinterrupts. Durch die zwei unterschiedlichen Typen von Tasks muß der Kontextwechsel für kooperierende und nichtkooperierende Tasks gesondert betrachtet werden. Die Laufzeit der Taskverwaltung für die durch Timerinterrupt in festem Zeitraster gestarteten kooperierenden Tasks liegt zwischen 1,63µs (327 clocks) und 1,97µs (271 clocks) (die Laufzeiten wurden auf einem mit 166MHz getakteten Prozessor gemessen). Die Zeiten sind davon abhängig, ob ein nichtkooperierendes Task unterbrochen wurde (maximaler Zeitverbrauch) oder nicht.
Die Laufzeit der Taskverwaltung für die nichtkooperierenden Tasks hat einen konstanten Zeitverbrauch für alle Tasks, der abhängig von der Taskanzahl ist. Ohne Compileroptimierung steigt der Zeitverbrauch mit der Taskanzahl linear an, mit Einsatz der Compileroptimierung steigt der Zeitverbrauch treppenförmig an.

4.4.2. Speicherplatzbedarf

Ein ohne eingebundene Nutzertasks compiliertes und gelinktes Beispielprogramm ist 33 kB groß. Wird auf Einsatz des Nachrichtensystems verzichtet, können 4 kB des Speicherbedarfs eingespart werden. Die Implementation der zur Kommunikation mit dem Host-PC vorgesehenen USB-Schnittstelle benötigt in ihrer ersten, nichtoptimierten Version 30 kB Speicher.

5. Kommunikation mit dem Host-PC

Die Kommunikation mit dem Host-PC geschieht über eine USB-Schnittstelle. Hardwareseitig besteht diese aus einer in Piggy-back-Technologie realisierten aufgesteckten Karte mit USB-Anschluß und USB-Chip PDIUSBD12 (Philips Semiconduktors). Softwareseitig ist die USB-Schnittstelle als nutzerinterruptgesteuerter nichtkooperierender Task implementiert.

6. Ablauf der Entwicklung

Für die Entwicklung wurde ein Experimentalaufbau unter Verwendung vorkonfektionierter CPU-Module in Betrieb genommen. Darauf wurden bzw. werden die Entwicklung und Erprobung des Echtzeitbetriebssystems und die Realisierung hardwarenaher Softwarekomponenten durchgeführt. Der Experimentalaufbau erweitert das Standalone-Sytem um Ein- und Ausgabefunktionen. (Siehe Abb. 3) Die Ein- und Ausgabevorgänge werden über den externen Bus des Prozessormoduls realisiert.
Eine Beschleunigung des Entwurfsablaufes und eine Verbesserung der Entwurfsqualität kann durch die Anwendung systematischer Entwurfs- und Testverfahren erreicht werden. Dazu gehören die Anwendung grafischer Darstellungsmittel für die Modellierung und Evaluierung von Hard- und Softwaresystemen sowie der Einsatz von integrierten Entwurfsumgebungen.

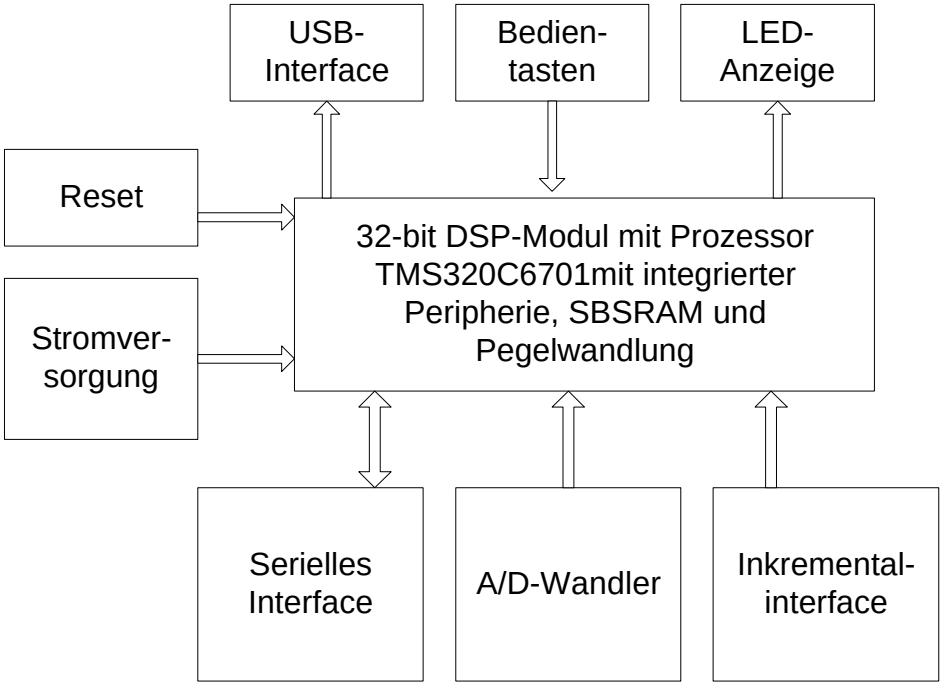

Abbildung 3: Blockschaltbild des Experimentalaufbaus

7. Ergebnisse und Ausblick

Kurz vor Beendigung der Projektlaufzeit ist das Echtzeitbetriebssystem eRTOS fertiggestellt und hat umfangreiche Tests absolviert. Die Implementation der USB-Schnittstelle wurde von einer bestehenden Version für D.Module C3x (mit Prozessoren einer älteren DSP-Familie von Texas Instruments) auf die von uns genutzten D.Module C6x portiert und wird zur Zeit in das Betriebssystem eingebunden. Für die Zukunft sind Tests mit neu entwickelten Meßwertverarbeitungsalgorithmen und Steuerfunktionen sowie die stärkere Einbeziehung systematischer Entwurfsverfahren anvisiert.

Quellenhinweise:

[1] Gerhardt, Uwe: Signalverarbeitung in der interferenzoptischen Meß- und Sensortechnik. Dissertation, TU Ilmenau 1997
[2] TMS320C6701 Floating-Point Digital Signal Processor: Data Sheet. Dokument-Code SPRS067D. Firmenschrift Texas Instruments, Februar 2000
[3] J. Hennessy, D. Patterson: Rechnerarchitektur. Analyse, Entwurf, Implementierung, Bewertung. Vieweg Verlag 1994

Autorenangaben:

Dipl.-Ing. Christian Uhle
Dr. Ing. Bernd Däne
Dr. Ing. Todor Vangelov
Prof. Dr. Ing. habil. Wolfgang Fengler
Fachgebiet Rechnerarchitektur
Fakultät für Informatik und Automatisierung
Technische Universität Ilmenau
Postfach 100565
98684 Ilmenau
Tel.: +49-3677-69-2825
Fax: +49-3677-69-1614
E-mail: {chris;bdaene;twangl;wfengler}@theoinf.tu-ilmenau.de