

Mit der Kraft mehrerer Kerne

Multicore-Unterstützung für SPS

Moderne Steuerungssysteme übernehmen immer mehr Aufgaben. Zuverlässigkeit und garantiertes Echtzeitverhalten dürfen darunter aber nicht leiden. Die Lösung liegt in leistungsfähigen Steuerungen mit Multicore-Prozessoren, bei denen die verschiedenen Kerne dedizierte Aufgaben übernehmen. Mit einer mehrkernfähigen Entwicklungsumgebung für speicherprogrammierbare Steuerungen nach dem IEC-61131-3-Standard lässt sich die Realisierung jetzt einfach umsetzen.

TEXT: Samuel Greising, 3S-Smart Software Solutions **BILDER:** 3S-Smart Software Solutions; iStock, Bet_Noire

Neue Steuerungsgenerationen enthalten immer häufiger CPUs mit zwei oder mehr Kernen. Steuerungshersteller versprechen sich durch die Aufteilung der Applikation in mehrere Kerne höhere Verfügbarkeit der einzelnen Komponenten, bessere Echtzeiteigenschaften unabhängiger Ausführungsstränge sowie eine bessere Performance des Gesamtsystems. In Mehrkernsystemen gibt es dann zum Beispiel je einen Kern für die SPS-Logik, einen für den Motion-Anteil und einen für die Feldbuskommunikation.

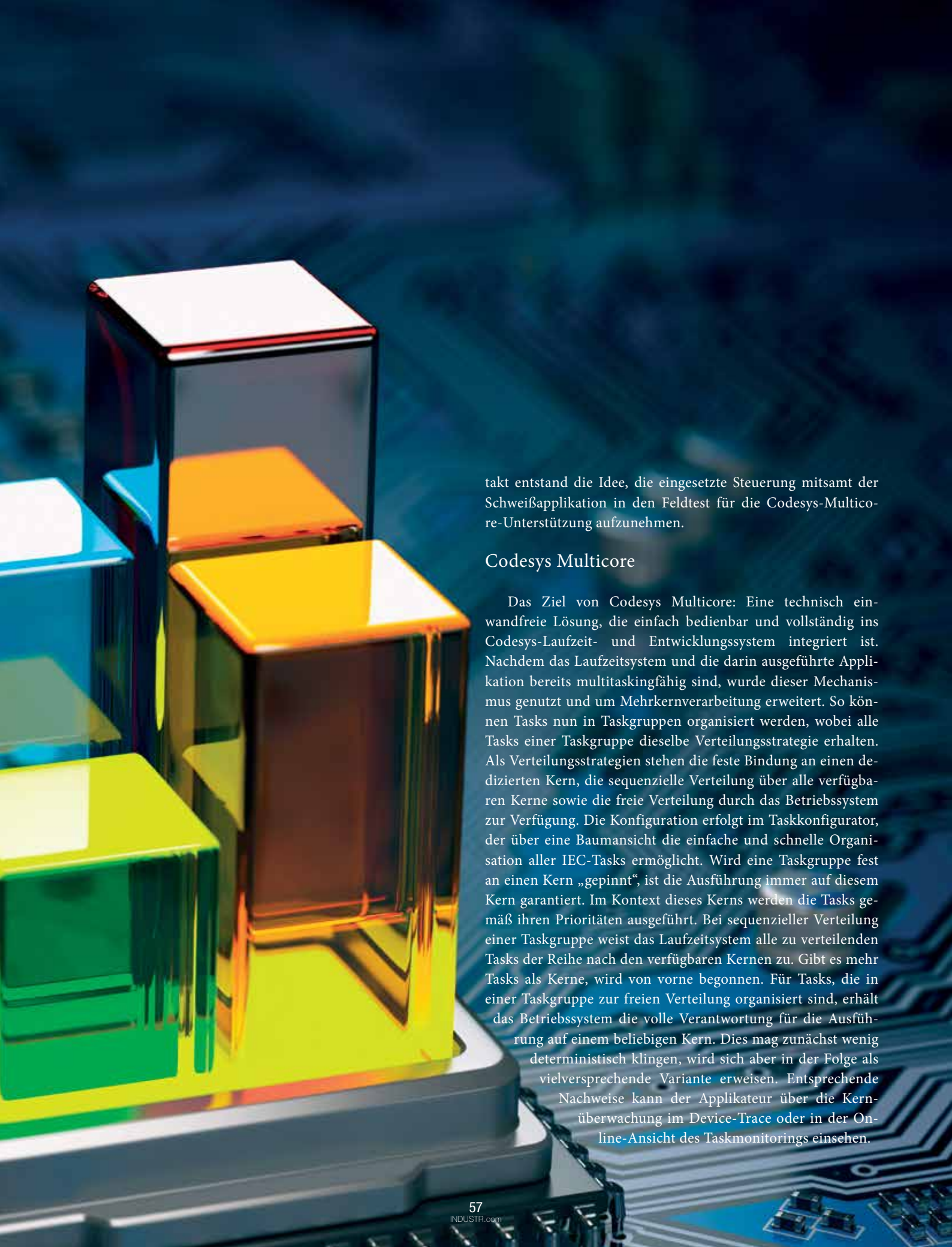
Typischerweise werden in diesen Steuerungen symmetrische Mehrkernprozessoren eingesetzt. Diese weisen eine identische Architektur der Kerne auf und teilen sich Ressourcen wie Hauptspeicher, Interrupts oder E/As. Dabei ermöglichen sie die unabhängige Ausführung von Code auf jedem Kern. Softwareseitig ist die Synchronisation und damit die Konsistenz der Daten und der Hardwarezugriffe zu gewährleisten. Bislang konnten Hersteller von Steuerungen mit Codesys-Laufzeitsystemen eine solche Aufteilung nur zwischen dem Laufzeitsystem und möglichen zusätzlichen Softwarekomponenten vornehmen. Eine Aufteilung des Codesys-Laufzeitsystems war bislang nicht möglich.

Probleme durch falsche Auslastung

Somit kann hier im Grunde gar nicht von „Mehrkernsystemen“ gesprochen werden, da das Codesys-Laufzeitsystem den allergrößten Teil der SPS-Funktionalität ausmacht und auch integrierte Zusatzfunktionen ausführt. Gerade bei Applikationen mit hohem Visualisierungsanteil, intensiven Motion- oder CNC-Berechnungen oder anderweitig rechenintensiven SPS-Aufgaben kamen die Applikateure bald an die Belastungsgrenzen der Systeme. Ein Beispiel hierfür ist die Applikation der Firma MS Ultraschall Technologie GmbH. Bei ihren Ultra-

schall-Schweißanlagen des Typs soniTOP maß sie weit über 90 Prozent konstante CPU-Auslastung. Dabei sind die mit Realtime-Betriebssystem und auf Intel x86 Quadcore (1,9 GHz) arbeitenden Steuerungen von TRsystems GmbH nicht gerade schwach ausgestattet.

Die Folgen dieser hohen Auslastung: Eine fast nicht zu bedienende grafische Benutzeroberfläche, Tasks mit niedriger Priorität, die fast nicht mehr zum Zuge kamen, und eine wackelige Echtzeitkommunikation via EtherCAT. Bereits seit einiger Zeit befassten sich TRsystems und MS Ultraschall mit den Möglichkeiten zur Ausnutzung aller verfügbaren Kerne. Denn während einer von vier Kernen am Anschlag arbeitete, waren die übrigen drei Kerne nahezu ungenutzt. So bereitete MS Ultraschall die Applikation für den Mehrkernbetrieb vor, teilte die verschiedenen Aufgaben der soniTOP-Anlage in separierbare Tasks auf und stellte Datenkonsistenz zwischen diesen Tasks sicher. Ohne die entsprechende Unterstützung des Codesys-Laufzeitsystems konnten hier jedoch keine entscheidenden Vorteile erzielt werden. Parallel dazu arbeitete man bei 3S-Smart Software Solutions bereits seit einiger Zeit an einer Multicore-Integration in das Codesys-Laufzeitsystem. Über den bereits bestehenden Kon-



takt entstand die Idee, die eingesetzte Steuerung mitsamt der Schweißapplikation in den Feldtest für die Codesys-Multicore-Unterstützung aufzunehmen.

Codesys Multicore

Das Ziel von Codesys Multicore: Eine technisch einwandfreie Lösung, die einfach bedienbar und vollständig ins Codesys-Laufzeit- und Entwicklungssystem integriert ist. Nachdem das Laufzeitsystem und die darin ausgeführte Applikation bereits multitaskingfähig sind, wurde dieser Mechanismus genutzt und um Mehrkernverarbeitung erweitert. So können Tasks nun in Taskgruppen organisiert werden, wobei alle Tasks einer Taskgruppe dieselbe Verteilungsstrategie erhalten. Als Verteilungsstrategien stehen die feste Bindung an einen dedizierten Kern, die sequenzielle Verteilung über alle verfügbaren Kerne sowie die freie Verteilung durch das Betriebssystem zur Verfügung. Die Konfiguration erfolgt im Taskkonfigurator, der über eine Baumansicht die einfache und schnelle Organisation aller IEC-Tasks ermöglicht. Wird eine Taskgruppe fest an einen Kern „gepinnt“, ist die Ausführung immer auf diesem Kern garantiert. Im Kontext dieses Kerns werden die Tasks gemäß ihren Prioritäten ausgeführt. Bei sequenzieller Verteilung einer Taskgruppe weist das Laufzeitsystem alle zu verteilenden Tasks der Reihe nach den verfügbaren Kernen zu. Gibt es mehr Tasks als Kerne, wird von vorne begonnen. Für Tasks, die in einer Taskgruppe zur freien Verteilung organisiert sind, erhält das Betriebssystem die volle Verantwortung für die Ausführung auf einem beliebigen Kern. Dies mag zunächst wenig deterministisch klingen, wird sich aber in der Folge als vielversprechende Variante erweisen. Entsprechende Nachweise kann der Applikateur über die Kernüberwachung im Device-Trace oder in der Online-Ansicht des Taskmonitorings einsehen.

Gruppenname	Core	Priorität
[-] Motion-Tasks	1	
EtherCAT_Master		0
[-] Realtime-Tasks	2	
CycleTask		1
HIGH		17
CrawlTask		31
[-] Low-Prio-Tasks	Sequentially Pinned	
LowerMainTask		18
LowerMainTaskB		19
LOW		20
AC_RMP_Task		28
[-] Visu-Tasks	Free Floating	
VISU_TASK_BACK		29
VISU_TASK		30

Codesys Multicore ermöglicht die feste Bindung von Tasks an einen dedizierten Kern, die sequenzielle Verteilung über alle verfügbaren Kerne sowie die freie Verteilung durch das Betriebssystem.

Die Multicore-Unterstützung im Codesys-Laufzeitsystem nutzt in erster Linie die gängigen und von den Betriebssystemen bereitgestellten Mehrkernfunktionen zur Verteilung, Synchronisierung und atomaren Datenverarbeitung. Über generische Schnittstellen im Laufzeitsystem werden diese abstrahiert und bieten somit die Basis für plattformunabhängige Verwaltung von Multicore-Betriebssystemen und -Prozessoren. Zudem ist die Kernverwaltung des Codesys-Laufzeitsystems in der Lage, jede beliebige Anzahl von Kernen zu organisieren – auch einen einzelnen. Dadurch gewährleistet Codesys auch weiterhin Plattformunabhängigkeit, sowohl was die Betriebssysteme, aber auch die CPUs anbelangt. Der Wechsel von einer Mehrkern-CPU zu einem einzelnen Kern und umgekehrt erfordert somit keinerlei Anpassungen der Applikation.

Für den Feldtest wurde zunächst das Laufzeitsystem der TRsystems-Steuerung um die Multicore-Unterstützung erweitert. Da VxWorks, Linux und Windows zu den Codesys-Standardplattformen zählen, fiel kein zusätzlicher Implementierungsaufwand an. Entsprechend schnell konnte die Multicore-Funktionalität nachgewiesen und mit kleinen Applikationen getestet werden. Im nächsten Schritt wurde die Ultraschall-Schweißapplikation in drei Phasen getestet, die Ergebnisse wurden miteinander verglichen. Dazu wurden die IEC-Tasks der Applikation zunächst in vier Taskgruppen unterteilt: Eine Gruppe für Echtzeittasks mit Motion und EtherCAT (jeweils 1 ms Taskzykluszeit), eine Gruppe mit einem hochprioritären Applikationstask (ebenfalls 1 ms Taskzykluszeit), eine Gruppe mit Tasks niedriger Priorität und größeren Taskzykluszeiten, sowie eine Taskgruppe für die Visualisierung.

Deutlicher Performancegewinn

Zusätzlich zu den explizit konfigurierten Tasks beziehungsweise Taskgruppen legt die Steuerung automatisch implizit Laufzeitsystem-Tasks für bestimmte Aufgaben an, zum Beispiel für die Kommunikation per OPC UA oder die Darstellung von Visualisierungselementen. Auch diese Tasks waren bislang ausschließlich auf einen CPU-Kern gebunden, werden aber mit der Multicore-Lösung jetzt automatisch auf die verfügbaren Cores verteilt. In der ersten Phase des Tests wurden daher die IEC-Tasks noch nicht verteilt. Die Messungen dieser Phase sollten den Performancevorteil aufzeigen, der allein durch die Verteilung von Laufzeitsystemtasks erreicht wird, ohne die Applikation zu ändern. Schlagartig sank die Auslastung des Hauptkerns von über 90 Prozent und pendelte sich bei guten 50 Prozent ein. Die zuvor unbedienbare Visualisierung war nun wieder reaktiv und konnte flüssig bedient werden, außerdem ging der Jitter aller Tasks deutlich zurück. Trotzdem war im Taskmonitoring deutlich zu sehen, dass die Kerne nicht gleichmäßig ausgelastet werden konnten.

Für die zweite Phase wurden die IEC-Taskgruppen daher so verteilt, wie man sich die beste Gesamtpformance ausgerechnet hatte: Die beiden Taskgruppen mit den Echtzeittasks wurden fest auf unterschiedliche Kerne gebunden, während die Gruppe mit den Tasks niedriger Priorität sowie die Visualisierungstasks frei verteilt wurden. Alle Tasks bekamen Prioritäten im Echtzeitbereich zugewiesen. Ergebnis: Ab diesem Moment wurden auch Animationen in der Visualisierung flüssig dargestellt. Der Jitter in allen Tasks war nun nochmals deutlich zu-

rückgegangen. Alleine im Taskmonitoring war erkennbar, auf welchen Kernen die Echtzeittasks ausgeführt wurden. Diese wiesen eine etwas erhöhte Last im Vergleich zu den anderen beiden Kernen auf.

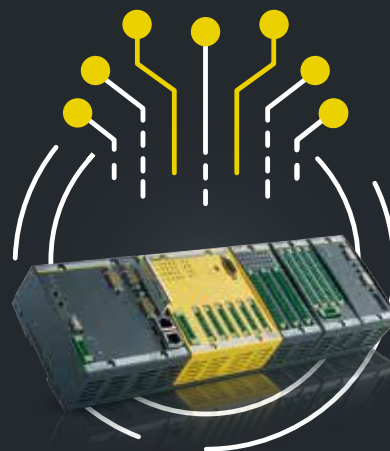
Die dritte Phase sah deshalb vor, dass alle Tasks dem Betriebssystem zur freien Verteilung überlassen werden – auch die Motion- und EtherCAT-Tasks. Im Ergebnis brachte diese Strategie die beste Gesamtperformance. Zum einen wurden die Kerne nun zwischen 10 und 30 Prozent und damit sehr gleichmäßig ausgelastet, zum anderen war auch beim Jitter nochmals eine deutliche Senkung zu sehen. Die Echtzeittasks wiesen nun fast keinen messbaren Jitter mehr auf – und dies, obwohl sie regelmäßig die Kerne und somit auch den gesamten Taskkontext wechseln mussten. Ein Langzeiteffekt, der im Rahmen der bisherigen Beobachtungen nur vorausgesagt werden kann: Durch die gleichmäßige Auslastung der CPU-Kerne wird die Steuerung thermisch weniger belastet und damit dauerhafter verfügbar sein.

SPS fit für künftige Anwendungen

Für korrekt ausprogrammierte Applikationen stellt Codesys den Geräteherstellern und Applikateuren mit der Multicore-Unterstützung ein mächtiges Werkzeug zur Verfügung, unter dessen voller Ausschöpfung eine völlig neue Qualität an Automati-

sierungsapplikationen möglich sein wird. Machine Learning, Anomalieerkennung und andere rechenintensive Operationen können mit Multicore auf SPS-Ebene gelöst werden, was bislang undenkbar oder nur mit starken Abstrichen umsetzbar war. Gerätehersteller und Codesys-Anwender sind damit für die großen Industrie 4.0-Themen bestens gerüstet. □

bachmann.



INTEGRIERT IN EINEM SYSTEM

Hochmodulare PAC/PLC

- Sicherheitssteuerung
- Visu/SCADA webbasiert
- Software Modulkonzept
- IT Security
- Flexible Programmierung
 - IEC 61131
 - C/C++
 - Modellbasiert (MATLAB®)
- SW Oszilloskop
- Motion Control
- Multi-Kommunikationslösungen
- I4.0 Lösungen
 - Flottenmanagement
 - Condition Monitoring
 - Fernwartung