

# So sieht zukunftsfähige Embedded-Softwareentwicklung aus

*Die Experten von MicroConsult diskutieren die Bedeutung von Softwarearchitekturen und Frameworks. Dabei wird das Thema Software Engineering aus verschiedenen Blickwinkeln beleuchtet.*

PETER SIWON \*



Bild: Siwon

tion bei großen Halbleiterherstellern. Sein besonderer Fokus liegt auf der Multicore-Technologie, die mehr und mehr bei den Mikrocontrollern an Bedeutung gewinnt.

## Das Potenzial von Multicore erschließen

„Mit dem Thema Multicore muss sich über kurz oder lang wohl jeder Embedded-Entwickler beschäftigen, wenn er nicht den Anschluss verlieren will. Die Frage ist weniger ob, sondern wann das erste Multicore-basierte Projekt realisiert wird. Multicore ist Realität und mehr Kerne und mehr Hersteller mit Multicore-Derivaten werden den Markt betreten. Eine Abstrahierung der Hardware ist dabei von großer Bedeutung, damit die Verteilung und Verwaltung von Software auf diese komplexe Hardware effizient möglich ist. Gerade hier gibt es aktuell allerdings noch Nachholbedarf“ so sein Tenor.

Doch was bedeutet „beschäftigen“? Marcus Gößler ist davon überzeugt, dass der effiziente Einsatz von Multicore ein sehr gutes Verständnis der Hardware-Architektur voraussetzt. Sie ist der Grundstein für die Entwicklung einer Software-Architektur, die das Leistungspotenzial von Multicore auch tatsächlich erschließt und die Unabhängigkeit und damit auch die Sicherheit gleichzeitig ablaufender Tasks gewährleistet. Maßnahmen, um diese Unabhängigkeit und Sicherheit auch überprüfen bzw. Fehler diagnostizieren zu können, kommen ins Spiel. Falls die einzelnen Kerne nicht völlig unabhängig operieren, sondern interagieren, müssen Maßnahmen ergriffen werden, um fehlerhafte gegenseitige Beeinflussungen zu vermeiden. Dies spielt üblicherweise bei Datenzugriffen eine Rolle, beispielsweise im Zusammenhang mit verwendeten Caches. Des Weiteren sind multicorespezifische Dinge wie Code- und Datenallokation zu betrachten. Die optimale Abstimmung von Hardwarearchitektur und Softwarearchitektur im Sinne von Leistungsgewinn und Sicherheit

*Alte, gewachsene Architektur hat durchaus ihren Reiz: Bei Software können wir uns Nostalgie in der Regel nicht leisten.*

**O**hne Zweifel wurde der Entwicklung von Embedded-Software noch nie so viel öffentliche Aufmerksamkeit geschenkt wie heute.

Internet of Things, Industrie 4.0, autonomes Fahren, Smart Building, Smart City, Smart Healthcare, etc., der Smartphone-Boom mit seinen Apps, aber leider auch der VW-Skandal weckten auch außerhalb der Fachmedien viel Aufmerksamkeit für Embedded-Software.

Embedded-Software ist allgegenwärtig und stellt einen wichtigen Faktor dar, wenn

es um Komfort, Sicherheit, Nachhaltigkeit und Innovation geht. Keiner kommt mehr an ihr vorbei. Sogar in der Bekleidungsindustrie bestimmt sie Funktion, Nutzen und Attraktivität der sogenannten Wearables.

Doch worauf kommt es bei all diesen Trends vor allem an, wenn es um das Software-Engineering geht? Wir fragten Thomas Batt, Marcus Gößler, Frank Listing und Remo Markgraf, die als Trainer und Berater für Embedded-Technologien seit vielen Jahren Kunden aus allen Industriebranchen begleiten.

Am besten fangen wir bei unserer Betrachtung ganz „unten“ an, beim Mikrocontroller: Marcus Gößler ist Spezialist für Mikrocontroller bei MicroConsult und sammelte vorher umfassende Erfahrungen als Field Application Engineer und Leiter der Field Applica-



\* Peter Siwon  
... ist für das Business Development der MicroConsult GmbH verantwortlich.

erfordert ein fundiertes Verständnis dieser beiden Aspekte.

Der Blick in die Gegenwart und die Zukunft zeigt deutlich, dass die Embedded-Welt und die IT-Welt miteinander verflochten sind und sich immer mehr verflechten werden. Embedded-Systeme werden zum Bestandteil des Internet of Things, das sich zu einem Internet of Everything entwickelt.

Bild: MicroConsult

### Interdisziplinäre Kooperation ist gefragt

Frank Listing betont, wie sehr klassische Embedded-Systeme (etwa Produktionsmaschinen) bereits heute mit Smartphones, Tablets, PCs und Servern interagieren. Er ist Spezialist für den Einsatz von PC- und Smartphone-Technologien im Embedded-Umfeld und war vorher selbst viele Jahre als Softwareentwickler und Softwarearchitekt in der Kommunikationsbranche tätig. Die Mensch-Maschine-Kommunikation läuft z.B. über Smartphones und Tablets, die sich mehr und mehr zu universellen Kommunikationsschnittstellen in allen Richtungen mausern. Doch der Komfort und die Vernetzung hat seinen Preis: Das Thema Security, also der Schutz gegen unautorisierte Zugriffe auf die Maschinen oder was immer über das Internet of Everything erreichbar sein wird, gewinnt massiv an Bedeutung.

Eine sehr viel höhere Kompetenz und engere Zusammenarbeit von Embedded- und IT-Spezialisten unter dem Aspekt Security ist nötig, um keine Einfallstore für Cyberkriminalität zu öffnen. Auch hier sind durchdachte Systemarchitekturen erforderlich, die diese Sicherheit gewährleisten und gleichzeitig genügend Flexibilität für Systemerweiterungen und Innovationen bieten.

Frank Listing betont außerdem, dass die Projektteams neben C/C++ auch C#, Java oder HTML5 mit JavaScript beherrschen müssen, um den neuen Anforderungen gerecht zu werden. Die Gräben zwischen IT und Embedded müssen schnellsten zugeschüttet werden, interdisziplinäre Projektarbeit ist gefragt.

### Softwarearchitektur-Design ist ein Muss

Thomas Batt, der bei MicroConsult das Thema Softwareengineering und Prozesse verantwortet, schlägt in dieselbe Kerbe. Er sieht, dass immer mehr Firmen den Nutzen einer definierten Softwarearchitektur zwar erkennen, doch der Weg aus dem Sumpf der Softwarealtlasten ist beschwerlich.

Die Art, wie Embedded-Software sehr oft in der Vergangenheit entwickelt wurde (oder sollte man sagen verwickelt wurde?) führt



**Ökosysteme:** Embedded-Software ist Bestandteil immer komplexerer Ökosysteme, die sich sehr dynamisch verändern.

nicht nur in eine Sackgasse sondern stellt ein wachsendes Sicherheitsrisiko dar. Die steigende Komplexität in Verbindung mit wachsenden Anforderungen an Safety (Betriebs-sicherheit) und Security (Systemschutz) stellt hohe Ansprüche an die Softwarearchitekturen und die Qualität der Implementierung. Dies wiederum spricht für den Einsatz von bewährten Softwarekomponenten (Be-

triebssysteme, Stacks, Filesysteme, etc.) und die Anwendung objektorientierter Methoden mit Hilfe von Hochsprachen wie C++.

Ziel muss es sein, der schleichenden Softwareerosion von Anfang an entgegenzuwirken. Aus der Sicht von Herrn Batt, der über langjährige Projekterfahrung in der Industrie verfügt, hat diese Erkenntnis auch dazu geführt, dass dem Requirements-Engineering

**universal debug engine®**

TriCore • PowerArchitecture  
RH850 • XC2000/XE166  
Cortex M/R/A • ARM7/9/11

**On Top Solutions for System Development of 16/32 Bit Microcontroller**

High SPEED  
ROBUST  
FLEXIBLE

Universal Debug Engine®  
Universal Access Device

Eclipse, C/C++ Compiler, Test Automation, Third Party Tools, COM Interface, CAN recorder, RTOS, JTAG, Trace

User-Specific Hardware / Evaluation Boards

www.pls-mc.com

**pls**  
Development Tools

**Softwarearchitektur:**

Der OPC-UA-Protokoll-stack ist ein Beispiel für eine offene und flexible Architektur zur Kommunikation im Umfeld von Industrie 4.0.

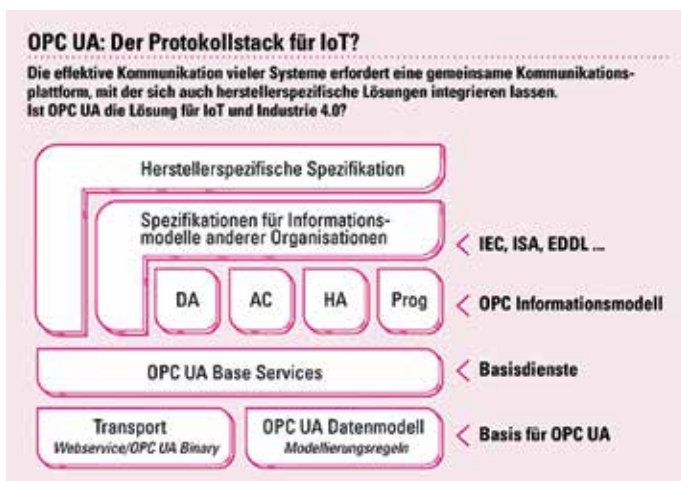


Bild: MicroConsult

Hersteller sollten schon jetzt auf eine passende Plattform, Software, Hardware und Tools, umsteigen, wenn sie schnell neue Gerätetechnologien auf den Markt bringen wollen. Eine erfolgversprechende disruptive Strategie erfordert die Wahl einer Plattform, die eine durchgängige und hohe Integrationstiefe von der Stange bietet, um sich auf die tatsächliche Innovation konzentrieren zu können ohne sich mit Toolintegration oder Detailprotokollen herumschlagen zu müssen.

Zum Glück haben Chip- und Toolhersteller die Zeichen der Zeit erkannt und bringen mehr und mehr Komplettumgebungen auf dem Markt, zum Beispiel Infineon, ST oder Renesas. Die Geräte-Hersteller, die diese Zeichen erkannt haben, werden sich schon in Kürze um die wenigen Entwickler, die Know-how-Träger in diesen neuen Themen sind, prügeln. Wer als Systemarchitekt, Entwickler oder Tester einer rosigen Zukunft entgegen sehen will, sollte sich schnell in die neuen Technologien einarbeiten, so Remo Markgraf.

Werfen wir gemeinsam mit Herrn Markgraf noch einen Blick auf die Ebene der Entwicklungsprozesse in der Embedded-Branche: „Jeder Entwickler leidet unter den ständigen 'Change-Requests' und dem steigenden Zeitdruck. Eine moderne Entwicklungsumgebung und Methodik sind wichtige Mittel, um diesen neuen Herausforderungen gewachsen zu sein.“ Die embedded Entwicklungsmethodik hinkte in der Vergangenheit der IT-Branche mehr als zehn Jahre hinterher, etwa beim Umstieg von Assembler auf C. Auch heute ist der zeitliche Versatz noch deutlich zu spüren, es sind noch schätzungsweise fünf Jahre. Momentan hat der zeitliche Versatz allerdings auch Vorteile: Embedded folgt nicht jedem Irrweg der IT, sondern kann sich die Rosinen herauspicken. Agile Frameworks wie Scrum oder Kanban, Extreme Programming wie Test Driven Development oder Pair Programming halten inzwischen auch in der Embedded-Welt Einzug.

Das Übernehmen aus der IT-Welt ist selten 1:1 möglich. Das komplexe Zusammenspiel aus Hardware und Software-Entwicklung und deren Test erfordert meist eine sehr individuelle Anpassung der Frameworks. Dies erfordert spezifisches Knowhow, um die Auswirkungen der Änderungen realistisch einschätzen zu können. Markgrafs Fazit: „Ohne agile Entwicklung und gut laufende Integration von der Anforderung über den Test bis zur Auslieferung sind die Themen IoT, Industrie 4.0 und SaaS schwer zu meistern.“ // FG

MicroConsult

und Management, sowie der Softwaremodellierung mehr Aufmerksamkeit geschenkt wird. Der Umstieg auf höhere Abstraktionsebenen in Kombination mit professionellem Softwareengineering ist unerlässlich, um der Komplexität Herr zu werden.

Ein weiteres Mittel, um Komplexität, erhöhte Sicherheitsanforderungen und schnelle Produktzyklen in den Griff zu bekommen sind Betriebssysteme, die den Applikationen der Hersteller zuarbeiten. Es kommt nicht von ungefähr, dass sich Embedded LINUX großer Beliebtheit erfreut und das liegt nicht nur daran, dass die Lizenzkosten entfallen. Es wird immer wichtiger, dass die Applikationsentwickler sich von den Details der Hardware lösen und auf wichtige Services und Schnittstellen zugreifen können, ohne sie selbst zu entwickeln. Hardware muss sich durch Hardwareabstraktion über das Betriebssystem schneller austauschen lassen. Hier gibt es im Bereich Multicore noch viel zu tun. Ideal wäre es, wenn dies auch für das Betriebssystem mittels eines Operating System Abstraction Layers (OSAL) möglich ist. Die System- und Softwarearchitektur sollte diese Aspekte jedenfalls berücksichtigen.

Remo Markgraf, Senior Management Consultant für Agile Methoden, Embedded-Programmierung und Test, war selbst lange Zeit

für Teams, Produkte und Projekte in der Kommunikationsbranche verantwortlich. Er sieht, dass Funktionalität und Services und damit auch Software immer mehr von den Embedded-Systemen ins Netz verlagert werden. Für den seltenen Fall offline zu sein, sorgt ein Notprogramm für eine Minimalfunktion.

**Plattformen müssen der Marktdynamik gewachsen sein**

„SaaS, 'Software as a Service' wird, wie C und C++, zur Selbstverständlichkeit auch für kleine Embedded Systeme,“ so seine Prognose. Es zeichnet sich ab, dass, so wie die Smartphone-Technologie unser Technikverständnis innerhalb von gut zehn Jahren revolutioniert hat, ein ähnlicher Evolutions-schritt durch das Thema „always-on“ der Geräte bevorsteht. Eine „Leap-Frog“ Strategie der kleinen Evolutionsschritte wird hier nicht zum Markterfolg führen, da neue Hersteller und solche, die den „disruptive“ Schritt gewagt haben, den Markt schneller besetzen können. Namhafte große und mittelständische Unternehmen werden hier möglicherweise auf der Strecke bleiben, analog etwa zu Nokia, die den Smartphone-Trend aus einer vermeintlich sicheren Marktposition verschlafen haben.

**Die Zukunft des Embedded-Designs**

Die rasante Entwicklung vom Internet of Things zum Internet of Everything macht eine enge Zusammenarbeit von Embedded und IT überlebenswichtig. Systeme und Software müssen den Ansprüchen dieser dynamischen Zukunft gewachsen sein, in der wir heute noch nicht wissen,

welche Applikation, Funktion oder Technologie morgen wettbewerbsentscheidend sein werden. Der professionelle Entwurf von flexiblen und sicheren System- und Softwarearchitekturen sowie Embedded-taugliche, agile Frameworks werden eine Schlüsselrolle spielen.