



Winterthur, 30. August 2011

swissT.meeting
Embedded Computing Conference



Key Note und Abstracts Stream 4

Stream 4

Entwicklung Tools

Developing Heterogeneous Embedded Systems, Simplified!

Mohamed Anas, MathWorks GmbH *

GCC: GNU Compiler Collection, ein strategisches Stück Software

Hans Buchmann, FHNW

Die Lösung im Fokus – Neue Strategien für komplexe Embedded Designs

Stephan Ahrends, National Instruments Europe; Christian Moser, National Instruments Switzerland Corp. Austin

Embedded-System-Entwicklung: Zeit gewinnen, Kosten senken, Komplexität beherrschen – mit Grafischer Programmierung von Mikroprozessoren

Marco Schmid, Schmid Elektronik/Engineering AG

UML statt «C» für Firmware

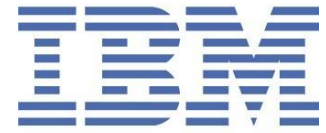
Walter van der Heiden, EVOCEAN GmbH

Open Source Tools für ARM Cortex-M3 basierte Mikrocontroller

Matthias Meier, FHNW

Embedded Software effizienter entwickeln dank der Verwendung eines Frameworks

Hugo Ziegler, CSA Engineering AG



Key-Note:

Research: The Art of Predicting and Shaping the Future

In der Research Division der IBM wurden und werden durch strukturierte Trendanalysen solide Vorhersagen hinsichtlich zukünftiger Technologien gemacht. Erstaunlich dabei ist, dass dabei mit guter Trefferquote etliche Jahre in die Zukunft geblickt werden kann.

Im Vortrag wird am konkreten Beispiel von einem modernen IBM Prozessor erklärt, wie dabei vorgegangen wird und welche Überlegungen bei den Vorhersagen eine Rolle spielen. Bei der anschliessenden Umsetzung in die Realität als kommerzielle IBM Computer Systeme spielt Embedded Computing in mehreren Bereichen eine wesentliche Rolle, sei es in der Sicherstellung von Funktionalität und Performance, sei es in der Bereitstellung von Plattformen für zugehörige Software-Entwicklung, oder aber während dem Vergleich und der Auswahl von unterschiedlichen Lösungsansätzen.

Referent: Dr. Martin Schmatz, IBM Research - Zürich

Developing Heterogeneous Embedded Systems, Simplified!

Modern embedded applications consist of complex structures and place the burden on source code. This presentation will use real world examples to emphasize the concept of Model-Based Design methodology addressing how such burden can be better managed.

Approaches to establishing a logical architecture to divide the model into components and the definition of the interfaces prior to such component design will also be discussed.

Attendees will see:

- an overview of graphical embedded application development tools
- production code generation in C, C++ and HDL
- approaches to achieving a significant reduction in verification effort
- how automatically generated code can be optimized and implemented on a target processor

Attendance is recommended for anyone involved in the following disciplines:

- Hardware and Software Engineering
- Test Engineering
- Project management

Referent: Dr. Mohamed Anas, Applications Engineering Group MathWorks GmbH

GCC: GNU Compiler Collection, ein strategisches Stück

Ohne die 'GNU Compiler Collection' GCC liessen sich viele Projekte nicht realisieren. Die GCC ist, gerade für die Entwicklung von eingebetteten Systemen, ein strategisches Stück Software.

Das Referat geht kurz auf die Geschichte der GCC ein und stellt sie in den Kontext der 'toolchain', dann werden ein paar prominente Eigenschaften der neuesten GCC Versionen vorgestellt.

Referent: Hans Buchmann, FHNW



Die Lösung im Fokus - Neue Strategien für komplexe Embedded Designs

Dieser Vortrag greift verschiedene Trends in der Embedded-Industrie auf.

Neue Embedded-Systeme müssen immer komplexere Funktionen zuverlässig ausführen. Gleichzeitig wird vom Produkte-Management eine kurze Markteinführungszeit sowie eine hohe Anpassungsfähigkeit an die verschiedensten Kundenwünsche bei gleichzeitiger Reduktion der Entwicklungskosten gefordert. Oft müssen spät im Entwicklungszyklus oder gar nach der Markteinführung, zusätzliche Funktionalitäten kostengünstig und schnell implementiert werden.

Die Teams zur Entwicklung komplexer Systeme werden immer interdisziplinärer – oft arbeiten z.B. Maschinenbauer und Physiker zusammen mit traditionellen Embedded-Entwicklern. Jeder dieser Experten muss in der Lage sein, sein Know How ins Design einzubringen und sich mit den anderen Teammitgliedern auszutauschen.

Hinzu kommt, dass die Technologie sich rapide weiterentwickelt und Technologieleader wie Intel, Xilinx, Freescale oder Windriver jedes Jahr neue Möglichkeiten liefern um Designs noch leistungsfähiger und noch ausgefeilter zu machen.

Im Vortrag wird aufgezeigt wie man hybride Designs mit einer integrierten Hardware- und Software-Tool-Chain adressieren und so Kostenvorteile erzielen kann.

Referenten: Stephan Ahrends, National Instruments Europe; Christian Noser, National Instruments Switzerland Corp. Austin

Embedded-System-Entwicklung: Zeit gewinnen, Kosten senken, Komplexität beherrschen - mit Grafischer Programmierung von Mikroprozessoren

Sensorsignale einfach auswerten. Mess- und Regel-Ideen schnell realisieren. Bedieneroberflächen (GUI) komfortabel erstellen. Frontends vernetzen. Der erfolgreiche Entwickler geht vom 1.Tag an produktiv auf Embedded HW/SW an die wichtigen Aufgaben heran. Im Vordergrund steht das Ziel/Ergebnis, die Details überlässt er den Tools (C-Code-Generator/Compiler). Mit "LabVIEW auf Mikroprozessoren" programmiert er funktionell in einer domänenspezifischen Sprache (DSL) der nächsten Generation (4GL). Interessant ist, dass auch bei hoher Abstraktion ähnliche Regeln und Gesetze wie beim traditionellen Vorgehen gelten, worauf im Vortrag eingegangen wird. Ein Erfahrungsbericht, wofür sich Grafische Embedded Anwendungsprogrammierung sehr gut eignet und wo die Grenzen liegen, rundet das Thema ab.

Referent: Marco Schmid, Schmid Elektronik/Engineering AG

UML statt "C" für Firmware

Obwohl es die UML (Unified Modelling Language) bereits seit über 15 Jahren gibt, kommt sie im Bereich Embedded Real-Time Systeme selten zum Einsatz; am ehesten noch für die Dokumentation. Aber genau bei diesen Anwendungen – welche hohe Anforderungen an Sicherheit, Zuverlässigkeit und Echtzeit stellen – wäre der Einsatz von UML zentral. Sowohl bei der Analyse, Architektur und dem Testing wie auch bei der Implementierung, Dokumentation und Kommunikation bietet UML grosse Vorteile.

Der Vortrag vergleicht UML mit der herkömmlichen Entwicklung in 'C'. Zur Illustration dient eine implementierte Lösung der Firma Willert Software Tools basierend auf IBM Rational Rhapsody, inklusive der durchgängigen Code Generierung aus dem Modell bis auf die Zielplattform.

Referent: Walter van der Heiden, EVOCEAN GmbH

Open Source Tools für ARM Cortex-M3 basierte Mikrocontroller

Etliche Silicon-Hersteller sind beim Ersatz Ihre alten Controller-Familien auf die günstige und leistungsfähige 32 Bit ARM Cortex-M3 Architektur umgestiegen - mit jeweils gleicher CPU- und Core-Architektur jedoch hersteller-spezifischen Peripherieeinheiten.

Nicht zuletzt ersparen sich die Hersteller dadurch auch die Entwicklung eigener Compiler- und Debugger-Tools. Statt dessen werden diese Tools von unabhängigen Software-Hersteller kommerziell angeboten, wobei einige "unter der Haube" wiederum den Open Source GNU C Compiler (GCC) einsetzen.

Dieser Vortrag zeigt, dass ein komfortables und produktives Arbeiten inkl. Remote-Debugging auch mit einer kostenlosen Open Source Lösung möglich ist – sowohl unter Windows wie unter Linux - sofern die kleineren und grösseren Installationshürden erst mal überwunden werden...

Referent: Matthias Meier, Fachhochschule Nordwstschweiz, Institut für Automation



Embedded Software effizienter entwickeln dank der Verwendung eines Frameworks

In diesem Referat wird dargelegt, wie durch Aufbau und Verwendung eines eigenen Frameworks die Effizienz der Entwicklung von Embedded-Software gesteigert werden kann.

Auf die Fragen wie: „Welche Funktionalitäten muss ein solches Framework bieten, was gehört alles rein?“, „Was muss bei dessen Entwurf berücksichtigt werden, dass es zum Erfolg führt?“, wird versucht, eine Antwort zu geben.

Es wird der Nutzen durch die **Wiederverwendung von Funktionalität** dargelegt und es werden auch die Vorteile bezüglich **Projektplanung** und **Projektmanagement** aufgezeigt.

Weiter werden die Aspekte wie **Skalierbarkeit, Testbarkeit, Erweiterbarkeit und Wartbarkeit** behandelt.

Auch das Thema **Unit-Tests** hat seinen Platz. Durch die Einbindung von Test-Frameworks wie z. B. GoogleTest wird es möglich, dass auch bei der Embedded-Software Entwicklung effizient **Unit-Tests** durchgeführt werden können.

Referent: Hugo Ziegler, CSA Engineering AG