ESS 2008 - TECHNICAL PROGRAMME

EMBEDDED SYSTEMS DEVELOPER CONFERENCE

Overview - Wednesday, 1ST October

Room	Gallery 1	Gallery 2	Gallery 18	Gallery 3
Theme	**Digital Convergence - Multicore Programming**	**Dependable Systems**	**Hands-On Classes**	**Hands-On Class**
9.30	**WELCOME ADDRESS**	**WELCOME ADDRESS**		
9.45	Tutorial 1.1 - TBA	Tutorial 2.1 – Dr Guillem Bernat, CEO, Rapita Systems **Timing matters: software optimization techniques for real-time systems**	**Embedded Linux**	**Beagle Board - Build Your Own** **This session is open to all attendees.** **Hands-on Beagle Boards are available as a limited allocation for two-day delegates only.**
10.45	Tutorial 1.2 – Stephen Blair-Chappell, Technical Consulting Engineer, Intel **Introduction to Multicore: Theory of Parallelism & Overview of implementation options**	Tutorial 2.2 – Dr. Michael J. Pont, CEO, TTE Systems **An introduction to Time-Triggered Architectures for reliable embedded systems**		
11.45	**COFFEE BREAK**			
12.00	Tutorial 1.3 – Mike Smith, Electrical and Computer Engineering, University of Calgary, Canada **Test Driven Development (TDD) of Reliable Systems across multiple embedded processor families and architectures**	Tutorial 2.3 – Trevor Jennings, Principal Engineer, Praxis High Integrity Systems **Static Verification for High-Integrity Software: Issues, Problems and Current Technologies**		
13.00	**LUNCH**	**LUNCH**	**LUNCH**	
13.45	Tutorial 1.4 – Stephen Blair-Chappell, Technical Consulting Engineer, Intel **Multicore Programming Hands-On Class** Special Note: Delegates will also have the opportunity to attend the hands-on class in the afternoon, where they can try out some of the techniques presented on multi-core target boards. Pre-registration for the afternoon hands-on class is essential to have access to the hands-on material which is strictly limited. The pre-registration process will take place directly after this morning session in the coffee breakout area. Spaces allocated on a first-come, first-served basis.	Tutorial 2.4 – Brian Hooper, Field Applications Engineer, LDRA **Benefits of Adopting Aerospace Development and Verification Standards in the Embedded Market**	**Advanced C Programming**	**Beagle Board - Build Your Own** **This session is open to all attendees.** **Hands-on Beagle Boards are available as a limited allocation for two-day delegates only.**
14.45		Tutorial 2.5 Dewi Daniels, Deputy Head of Consultancy, Silver Software **RTCA DO-178C / EUROCAE ED-12C — A Progress Report**		
15.45		Tutorial 2.6 Glennan Carnie, Technical Manager, Feabhas **Use Case Analysis - the baker's dozen of use cases**		
16.45	**DRINKS RECEPTION**			

ESS 2008 - TECHNICAL PROGRAMME

EMBEDDED SYSTEMS DEVELOPER CONFERENCE

Overview - Thursday, 2nd October

Room	Gallery 1	Gallery 2	Gallery 18
Theme	**Digital Convergence**	**Dependable Systems – Code Generation**	**Hands-On Classes**
9.30	**WELCOME ADDRESS**	**WELCOME ADDRESS**	
9.45	Tutorial 3.1 – Tim Wilmshurst, Head of Electronics, University of Derby **Power Consumption Mechanisms in Embedded Systems – the Hardware Issues**	Tutorial 4.1 – Matthew Fowler, CEO, New Technology/enterprise (NT/e) **Code Generation - The Fundamentals**	**Concurrency in C++**
10.45	Tutorial 3.2 - Abi Nourai, European Technical Director, Open Kernel Labs **How Secure is Your Embedded Design?**	Tutorial 4.2 – Speaker TBA **MDA Explained. (Describing the OMG's MDA approach from an embedded perspective.)**	
11.45	**COFFEE BREAK**		
12.00	Tutorial 3.3 - Professor Geoff Lawday, Tektronix Professorial Chair in Measurement, Buckinghamshire New University **Low Power Issues on High-Speed Comms Busses**	Tutorial 4.3 – Karsten Thoms, Software Architect, Itemis **Introduction to the Eclipse Modelling Project**	
13.00	**LUNCH**	**LUNCH**	**LUNCH**
13.45	Tutorial 3.4 - Dr Andrew Eliasz, Founder, First Technology Transfer (FTT) **Ad Hoc Sensor Networking - Concepts, Strategies and Systems**	Tutorial 4.4 – Karsten Thoms, Software Architect, Itemis **Model it your way - Efficient modelling using Domain Specific Languages**	**Embedded Linux**
14.45	Tutorial 3.5 – Ian Marsden, CTO, Eseye **Making the Machine2Enterprise Cellular Enabled**	Tutorial 4.5 – Axel Terfloth, Itemis **Testing Code Generators**	
15.45	Tutorial 3.6 – Tom Cooksey, Software Engineer, Nokia-Qt Software **Using OpenGL ES APIs on Embedded Linux**	**PANEL SESSION** Moderator: Andrew Watson, Technical Director, OMG **Discussion: 10 Years from now nobody will write code by hand**	
16.45	**DRINKS RECEPTION**		

ESS 2008 - TECHNICAL PROGRAMME

EMBEDDED SYSTEMS DEVELOPER CONFERENCE

Wednesday, 1ST October

Room:
Gallery 1

Digital Convergence

- Embedded Multicore Processor Programming
- Design for Low Power Consumption
- Hardware and Software Interoperability
- Wireless System Development

9.30 **WELCOME ADDRESS**

09.45
Tutorial 1.1 – TBA

10.45
Tutorial 1.2 -

Stephen Blair-Chappell, Technical Consulting Engineer, Intel
Introduction to Multi-Core Programming: Theory of Parallelism & Overview of Implementation Options

With the emergence of multi-core embedded CPUs and microcontrollers, comes the new challenge of how to program these devices to take advantage of the extra cores. In this presentation we show how existing programs can be modified to take advantage of the new multi-core architecture; we will show how to use different programming paradigms to implement multithreading in a program; we explain some of the common pitfalls that occur when writing for multi-core, and how best to avoid and detect these pitfalls; we also show how to tune and optimise multi-core programs.
The presentation will contain real code examples, and a 'live' demo of the complete development cycle showing how to spot the best place to insert parallelism; how to implement the parallelism; how to check for correctness, and how to tune a parallel program.
If you are considering moving your designs to multi-core, then you should attend this session.

Bio:
Stephen is a Technical Consulting Engineer at Intel, and has worked in the *Intel Compiler Lab* for the last 10 years. He is a regular speaker at technical conferences in Europe and the US.
Prior to joining Intel, Stephen worked as a lecturer at the University of Central England, specializing in Software Engineering and Embedded Systems. As an academic he developed and delivered CPU architecture programming courses for a number of companies including AMD and CAD-UL.
Outside work, Stephen enjoys playing the pipe organ, and is an accomplished musical instrument restorer.

Special Note: Delegates will also have the opportunity to attend the hands-on class in the afternoon, where they can try out some of the techniques presented on multi-core target boards.
Pre-registration for the afternoon hands-on class is essential to have access to the hands-on material which is strictly limited.
The pre-registration process will take place directly after this morning session in the coffee breakout area. Spaces allocated on a first-come, first-served basis.

Related Information Links:
www.intel.com/software
www.intel.com/technology/advanced_comm/multicore.htm
www.intel.com/go/parallel

11.45
COFFEE BREAK

12.00
Tutorial 1.3 -

Mike Smith, Electrical and Computer Engineering, Schulich School of Engineering, University of Calgary, Canada
Test Driven Development (TDD) of Reliable Systems across multiple embedded processor families and architectures.

A key element recognized as causing failures is associated with very limited training in, and minimal support for, testing of software and hardware combinations. In the desk top business world, Test Driven Development (TDD) has proven an important process for reducing the introduction of defects and lowering production and post-production costs. We have developed Embedded_UnitTest++ to gain the same pro-active, defect inhibiting advantages when developing reliable embedded systems. This testing framework is designed to work across a variety of processor families and architectures to support reliable new product development and existing product upgrades possibly combined with processor migration.
Our talk will provide illustrations of the testing framework used to pro-actively identify a variety of common embedded issues. We include a recent industrial case study and current research results on our hardware assisted testing methods to avoid the code bloat and associated performance distortions of software instrumented approaches to identify data races in multithreaded systems and test suite coverage issues.

Bio:
Mike Smith graduated from the University of Hull and emigrated to Canada to undertake a PhD involving magnetic resonance and high speed air turbines. He later undertook postgraduate work developing magnetic resonance hardware and software for breast cancer detection. After teaching physics, mathematics and computing at a high school, he joined the Department of Electrical and Computer Engineering at the University of Calgary, Canada. He is involved in collaborative work in developing biomedical engineering algorithms for magnetic resonance stroke studies; and with the University of Alberta, Edmonton in identifying reliable software practices to move these algorithms to embedded systems. He has been Analog Devices University Ambassador since 2001.

Related Information Links:
http://enel.ucalgary.ca/People/Smith/embeddedTDD/EmbeddedSystemsShowBirmingham2008.htm

13.00
LUNCH

13.45
Tutorial 1.4 -

Hands-On Class

How to Program for Multi-Core
Stephen Blair- Chappell, Technical Consulting Engineer, Intel

Description
With the introduction of multi-core processors comes the new challenge of how to make use of this extra processing power. Traditionally the majority of programmers wrote their programs without implementing any threading or parallelism. In this seminar you will learn how to implement parallelism in existing legacy code, what pitfalls to look out for, and how to profile the resulting parallel program.

The seminar is aimed that the professional developer who has not had any experience in writing parallel programs. The seminar will include extensive hands-on sessions, where the participants will take an existing serial program and turn it into a parallel program using a variety of techniques.

Pre-requisites
Attendees must be familiar with the C programming language, and be able to write moderately difficult C programs.

Details

Session 1: **Implementing Parallelism Using OpenMP (90 minutes)**
Introduction to the OpenMP standard
Loop Parallelism
Task Parallelism
The OpenMP Runtime Environment
Includes extensive hands-on session

Session 2: **Checking for correctness (30 minutes)**
How to find Data Races, Deadlocks and Contention issues.
Includes extensive hands-on session

Session 3: **Profiling and Tuning Parallel Programs (60 minutes)**
Profiling Parallel applications
Measuring Synchronisation overhead
Load Balancing
Includes extensive hands-on session

Related Information Links:
www.intel.com/software
www.intel.com/technology/advanced_comm/multicore.htm
www.intel.com/go/parallel

17.00
CLOSE

WEDS, 1.10.08

Room:
Gallery 2

Dependable Systems

- Development and Verification Standards in the Embedded Market
- Developing for System and Software Security
- Testing Hardware for Reliability
- Software Development for Critical Systems

9.30 **WELCOME**

9.45
Tutorial 2.1 -

Dr Guillem Bernat, CEO, Rapita Systems
Timing matters: software optimization techniques for real-time systems

"How long does my software take to run…and how can I reduce it?" are key issues for embedded engineers designing reliable systems. Understanding, verifying, and improving the timing performance of their real-time products gives successful companies in the avionics, telecommunications, space, and automotive electronics industries a key competitive edge.
Today, software timing analysis doesn't have to be guesswork. This presentation addresses two key aspects of real-time systems performance:
(1) how to gain a clear, detailed, and accurate understanding of the execution time behaviour of real-time embedded software,
(2) how to target optimisation effort precisely where it will have the maximum benefit in improving system timing behaviour, for the minimum cost.
This presentation will cover key aspects of real-time systems: finding worst case execution times (WCET) and the worst case path, and why worst case optimization is not the same as average execution optimization.
Practical issues of looking for the worst-case "hot-spots", identifying timing bugs and verifying optimisation opportunities will be explained with examples of worst case optimizations.

Bio:
Dr Guillem Bernat received his PhD in Computer Science from the Universitat de les Illes Balears in 1998. Since then, he has lectured extensively in real-time systems and is a frequent speaker at international conferences; renown for both his interesting presentations and enthusiastic style. Guillem is acknowledged as one of the world's leading experts on worst-case execution time analysis. He is a major contributor to the theory of probabilistic timing analysis for real-time systems and the architect of the RapiTime tool-set. Guillem is the founder and CEO of Rapita Systems Ltd., a spin-off company from the University of York, transferring measurement based worst-case execution time analysis technology into commercial practice.

Related Information Links:

10.45
Tutorial 2.2 –

Dr. Michael J. Pont, CEO, TTE Systems
An introduction to time-triggered architectures for reliable embedded systems

In a time-triggered embedded system, we have **one** (and only one) interrupt enabled. This interrupt is usually linked to a timer, which will generate "ticks": these ticks will, in turn, drive an appropriate (often very simple) operating system.

Time-triggered architectures are widely used in safety-related systems because they are known to provide highly predictable behaviour, which in turn reduces testing, maintenance and (where relevant) certification costs. During this talk it will be argue that TT architectures can be (and should be) used more widely. Numerous examples of the use of TT architectures will be presented. Strengths and weaknesses of the TT approach will be considered.

The talk will not try to argue that a TT approach is a perfect match for all embedded systems. However, the talk will conclude by suggesting that – as you start your next project – you should ask yourself: "Should we use a TT architecture this time?" By the end of this talk, you may be surprised how often the answer is: "Yes".

Bio:
Michael J. Pont is CEO of TTE Systems Ltd and Head of the Embedded Systems Laboratory at the University of Leicester. Since the mid 1990s, Michael has worked with an internationally-recognised research team to develop tools and techniques which support the rapid development of reliable embedded systems. This research work has resulted in a number of patent applications and the creation of the RapidiTTy™ tool family, which is now being further developed and sold by TTE Systems Ltd. Michael is author or co-author of more than 100 technical papers, and is the author of three books (including "Patterns for Time-Triggered Embedded Systems" and "Embedded C").

Related Information Links:
1. Introduction to TT architectures: http://www.tte-systems.com/technology/basics
2. Book about TT architectures (1000 pages, free download) plus detailed code examples (also free download): http://www.tte-systems.com/books/pttes

11.45
COFFEE BREAK

12.00
Tutorial 2.3 -

Trevor Jennings MBCS MIET CEng, Principal Engineer, Praxis High Integrity Systems
Static Verification for High-Integrity Software: Issues, Problems and Current Technologies

This session will present a tour of static verification technology and its use in high-integrity embedded software development.
The talk opens with a discussion of the "big five" goals for a static verification system (soundness, completeness, efficiency, expressive power, and modularity), and how these pose major challenges for language designers and tool vendors.
The various approaches and analysis styles will then be considered, focussing on the pros and cons of each, including retrospective vs. constructive analysis, "whole" languages vs subsets, and the use of so-called annotations or contracts.
This section will be illustrated with examples of major languages and tools in each category including MISRA C, SPARK/Ada, and Real-Time Java. The session will finally cover how users can evaluate and select languages and static verification tools, and how such technology can contribute to the evidence-based assurance required by contemporary engineering approaches.

Bio:
Trevor is a Professional Technical Consultant and experienced Engineer who has worked on a wide range of projects and technologies in the Electronics, Automotive and Communications industries. He is a principal engineer with Praxis High Integrity Systems where he has worked on a variety of high-integrity engineering projects.
While he was a research fellow at Southampton University, Trevor co-authored the first design of the SPARK language definition, and remains a full-time member of the SPARK R&D team at Praxis to this day.

Related Information Links:
www.sparkada.com

13.00

LUNCH

13.45
Tutorial 2.4 -

Brian Hooper, Field Applications Engineer, LDRA
The Benefits of Adopting Aerospace Development and Verification Standards in the Embedded Market

We will look at the increasing uptake of rigorous standards by non-aerospace industries. What are the key challenges being faced as companies evolve to adopt new standards? Greater formality in development processes, conforming to coding standards, and demonstrating verification of components via code coverage – all represent a step up in effort and approach. It therefore makes sense to analyse how aerospace projects have been working to meet certification criteria over the last 30 years.
An ever-increasing reliance on software control has meant that many companies from non-aerospace business sectors, not traditionally requiring sophisticated software development processes, now find themselves compelled to undertake safety-critical and safety-related analysis and testing. Consider the modern motor car equipped with all manner of safety systems, such as anti-lock braking and traction control, each managed by software running on a networked set of processors; the failure of any of these systems will be a major safety concern and could even lead to recalls and lawsuits.

Bio:
Brian Hooper is a Field Applications Engineer at LDRA Ltd. He graduated from Brunel University in 1989 before beginning a career in software development, first at GEC Marconi, then at several large defence contractors across Europe. In 1999, Brian joined Rational Software to support their successful Aerospace & Defence business, making the most of the software development lifecycle skills and experience he had picked up in industry. Brian has been with LDRA since 2007 and specialises in requirements management, requirements traceability, process and development best practices.

14.45
Tutorial 2.5

Dewi Daniels, Deputy Head of Consultancy, Silver Software
RTCA DO-178C / EUROCAE ED-12C — A Progress Report

DO-178B/ED-12B provides guidance for the production of software for airborne systems and equipment. Almost without exception, all the software on aircraft by manufacturers such as Airbus and Boeing has to be written to satisfy the objectives of DO-178B/ED-12B. There have been many advances in software engineering since DO-178B/ED-12B was published in 1992. New technologies have come into widespread use, including object-oriented technology (e.g. UML), model-based development (e.g. Simulink and SCADE) and formal methods (e.g. Z). DO-178B/ED-12B is being updated to DO-178C/ED-12C to fix ambiguities that have been found in DO-178B/ED-12B, as well as to address these new technologies. This update is being carried out by WG-71/SC-205, a joint EUROCAE Working Group and RTCA Special Committee. This presentation will update you on the progress being made in preparing DO-178C/ED-12C, will let you know what kind of changes to expect, and explain how DO-178C/ED-12C will affect your airborne software development processes.

Bio:
Eur Ing Dewi Daniels is a Senior Technical Consultant at Silver Software, a specialist safety-critical software company. Dewi is a hands-on software developer with 27 years experience of developing mission critical and safety-critical software. He was one of the developers of the SPARK Examiner, a toolset that is widely used for safety-critical software development in Ada. He has worked on aircraft programmes including Airbus A340, A380, MRTT and Boeing 787. Dewi is European Chair of SG2.1, one of the subgroups preparing DO-178C/ED-12C.

Related Information Links:
Silver Software – www.silver-software.net
SC-205 website – http://ultra.pr.erau.edu/SCAS/

15.45

Tutorial 2.6

Glennan Carnie, Technical Manager, Feabhas
Use Case Analysis - the baker's dozen of use cases

Use cases have become a core part of the requirements analyst's arsenal. Used well they can bring dramatic increases in customer satisfaction and a whole host of other subtle benefits to software development.
The use case itself is very simple in concept: describe the functionality of the system in terms of interactions between the system and its external interactors. The focus of the use case is system usage, from an external perspective.
Despite this apparent simplicity, requirements analysts frequently struggle to write coherent, consistent use cases that can be used to facilitate development. Often, the use case analysis becomes an exercise in confusion, incomprehension and the dreaded 'analysis paralysis'.
This presentation aims to aid use case writers by presenting a set of rules to follow when performing use case analysis. The rules are designed to avoid common pitfalls in the analysis process and lead to a much more coherent set of requirements.

Bio:
Glennan Carnie is Technical Manager with Feabhas ltd. He has spent most of his career in high integrity systems, from aerospace to banking. Glennan specialises in software architecture, requirements analysis and software development processes. In addition to these, he also lectures in UML, SysML, C++ and C for Feabhas.

Related Information Links:
www.feabhas.com

16.45
DRINKS RECEPTION

WEDS, 1.10.08

Room:
Gallery 3

Hands-On Class

Presented by Feabhas Ltd

9.45 - 12.45

Embedded Linux: a practical approach

Workshop Description:
This session shows how to build a working embedded Linux system, using a Power PC development board as the target. Topics covered include cross-development tools, kernel configuration and compilation and the essential components of a root file system. The combination of theory and practice will give delegates an insight into the working of Linux in embedded environments. The workshop is 50% practical and 50% theory.

Workshop Objectives:

- To get real-world exposure to embedded Linux
- To develop an application to run on an embedded Linux system.

Delegates will learn:

- How to configure a standard Linux kernel for use in a cross development system.
- The steps to write, compile and download an embedded Linux application with real hardware.

Who Should Attend:

- Engineers wishing to assess the suitability of Linux for their next application.
- Software engineers who are developing applications for embedded Linux.

Pre-Requisites

- Good 'C' programming skills
- General knowledge of an RTOS or embedded operating systems
- Knowledge of Linux/Unix commands (e.g. ls, cat, cp) will be useful, but not essential

Duration:
3 hours.

Course Materials:
Student workbook.

Course Workshop:
The course presents embedded concepts applied to Linux, using an ARM development board as the target (ConnectCore Wi-9C Digi JumpStart Kit for Linux). The host development system will be a standard PC running desktop Linux. We use the target as an example of a simple embedded system which can control hardware via a simple digital I/O interface.

Related Information Links:

16.45
DRINKS RECEPTION

WEDS, 1.10.08

Room:
Gallery 3

Hands-On Class

Presented by Feabhas Ltd

13.45 – 16.45

Advanced C Programming

Workshop Description:
This session will explore one of the trickier aspects of the C language – namely pointers. Our Advanced C Course (AC401), from which the workshop material is drawn, has shown that even quite experienced C programmers can still have difficulty with some pointer concepts – particularly their relationship with arrays and their use as function pointers. The workshop is 50% theory and 50% practical.

Workshop Objectives:
- To re-visit C pointers in general and to consolidate knowledge of their more esoteric uses
- To complete the coding of a real-time system, using techniques based upon the presented material, and to get the system to run

Delegates will learn:
- The relationship between pointers and arrays, and when to use which.
- The declaration and uses of function pointers.
- The dangers inherent in using pointers, and how to circumvent disaster

Who Should Attend:
- Engineers who have already used C in embedded systems but have not necessarily exploited the language (especially pointers) to the full.
- C programmers who would like a new perspective on how to use pointers effectively
- C programmers who would like to test the depth of their C knowledge

Pre-Requisites:
- Moderate 'C' programming skills
- General familiarity with embedded systems

Duration:
2½ Hours

Course Materials:
Student workbook.

Practical work:
The practical work will involve cross-development on a PC for an embedded system based on an ARM processor. Participants may test and run their programs using a simulated environment provided as part of the cross-development toolset.

Related Information Links:

16.45
DRINKS RECEPTION

ESS 2008 - TECHNICAL PROGRAMME

EMBEDDED SYSTEMS DEVELOPER CONFERENCE

Thursday, 2nd October

Room:
Gallery 1

Digital Convergence

- Embedded Multicore Processor Programming
- Design for Low Power Consumption
- Platforms for Hardware and Software Interoperability
- Wireless System Development

9.30 **WELCOME ADDRESS**

09.45
Tutorial 3.1 –

Tim Wilmshurst, Head of Electronics, University of Derby
Power Consumption Mechanisms in Embedded Systems – the Hardware Issues

Achieving low power consumption is one of the main targets for designers of modern embedded systems. It is the hardware which consumes the power, but software strategies have a major impact. This presentation explains clearly what the power consumption mechanisms are in embedded systems, how microcontrollers are designed to make best uses of power, and how power consumption can be minimised.
Review of power in the electrical context – Volts, Amps and Watts; how this applies in the embedded system.
Power consumption mechanisms in CMOS, dependence on clock frequency, supply voltage, logic levels and transition rates.
Features of the low power microcontroller: clock frequency range, supply voltage range, current consumption characteristics, clock cycles per instruction.
Sleep, Idle and similar Modes: what the modes do, what can be done within them, how to exit from them.
Minimising power consumption through clock frequency manipulation: use of phase locked loop, clock frequency division, power-down of unused peripherals.

Bio:
Currently Head of Electronics at University of Derby. 11 years Electronics Design Engineer/ Senior Electronics Design Engineer, Engineering Department, University of Cambridge. Experienced lecturer and presenter, including at Universities of Cambridge and Derby.

Related Information Links:

10.45
Tutorial 3.2 –

Abi Nourai, European Technical Director, Open Kernel Labs
How Secure is Your Embedded Design?

The paper will discuss both current and newly-emerging threats to embedded system security, with particular reference to examples taken from intelligent mobile devices such as Smartphones. The paper will then describe an architectural approach which allows the embedded system developer to implement device security at an appropriate level to meet the needs of their particular application, whilst keeping the impact on system performance to a minimum. It will then continue by demonstrating how to use an

embedded CPU with an MMU to further enhance the security of all the software components of an embedded system environment.

- How to run large application OS'es such as Linux, Symbian and WindowsCE safely and securely;
- Why device drivers are a particularly difficult security issue and what to do about it;
- How the use of a secure inter-process communication mechanism to implement "Segregated Capabilities" provides an additional line of defence against intrusion and/or unstable code;
- Using multiple secure partitions within a system to provide the required level of security for particular applications.

Bio:
Prior to joining OK Labs, Abi Nourai was Project and Team Lead with Dexion Integrated Systems, developing and delivering large software and hardware systems. Abi has a long history working with the OK Labs L4 microkernel/virtualization technology, going back to his undergraduate studies at the University of New South Wales where he received a double degree in Pure Maths and Computer Engineering (with First Class Honors and the University Medal). Abi joined OK Labs as Engineering Director last year, where he resurrected his passion for L4 and its applications. Recently, Abi has relocated to the OK European HQ in Paris as Technical Director for Europe.

Related Information Links:
www.ok-labs.com

11.45
COFFEE BREAK

12.00
Tutorial 3.3 –

Professor Geoff Lawday, Tektronix Professorial Chair in Measurement, Buckinghamshire New University
Implementing the new generation of low-power, high-speed, serial buses

Media convergence is technically demanding and poses many design and implementation challenges; parallel computing with multiple core processors is subject to Amdahl's Law and high-performance portability is dependent on reducing the watts per instruction. What is more, the buses remain the principal factor in determining the performance of embedded systems. While delivering orders of magnitude in improved bandwidth per pin the new high-speed serial buses are not transparent and require expert design if a low-power, high-speed, interconnect is to be achieved.
The new generation of high-speed serial buses have edge rates with radio frequency (RF) parameters and analogue characteristics, two subjects that many digital engineers have limited experience of. This presentation will explain in practical terms the trade-offs in power and performance in modern bus design and implementation, along with the exciting new integrated serial bus simulation and real-time, design and test paradigms.

Bio:
Dr Geoff Lawday holds the Tektronix Professorial Chair in Measurement at Buckinghamshire New University where he delivers lectures on high-performance computing and post-graduate mobile computing courses. He co-authored the latest book A Signal Integrity Engineer's Companion in the Prentice Hall best selling SI series and is widely respected for his technical papers and signal integrity workshops.

Related Information Links:
Book: http://www.amazon.co.uk/Signal-Integrity-Engineers-Companion-Measurement/dp/0131860062
Paper: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1271882

13.00
LUNCH

13.45
Tutorial 3.4 –

Dr Andrew Eliasz, Founder, First Technology Transfer (FTT)
Ad Hoc Sensor Networking - Concepts, Strategies and Systems

The development of powerful, low-cost, low-power microcontrollers with or without on-chip radio communications capabilities has opened up a world of applications based around communicating collections of adaptable (smart) sensors. More complex ad hoc networks may be made up of both stationary and mobile sensors and the sensors themselves may have GPS capabilities.

This presentation will look at issues in developing ad hoc network based applications, such as;

- power consumption and power source issues
- whether to use a general purpose processor or a multi-processor core
- interoperability issues - proprietary or standardised protocols
- portability issues - will the software be targeted at a specific platform or not
- networking and security issues - how are the various nodes discover one another
- re-configurability issues - such as the ability of a network to repair itself
- link layer and physical layer issues of the underlying wireless networks
- system testing - both at the hardware and software level
- system modelling - to test out various scenarios before actual system deployment

Bio:
Dr Andrew Eliasz established FTT with a view to providing specialist, innovative and highly tailored technical training and consultancy to organisations developing both traditional IT and embedded systems applications.
Dr Eliasz has over 30 years experience of delivering training courses such as C, C++, Unix, Networking, Perl, advanced databases and systems analysis and design. In that time he has also been involved in various consultancy projects including, work on implementation of industrial protocols such as the HART protocol and on the implementation of firmware for fuel cell sensors.

Related Information Links:
http://www.ftt.co.uk

14.45
Tutorial 3.5 –

Ian Marsden, CTO, Eseye
Making the Machine2Enterprise Cellular Enabled

With the ubiquity of cellular networks, the cellular enablement of remote products is becoming increasingly viable both on technical and cost grounds. However such a system has a multitude of different components to integrate successfully before a product is ready to go to market.
This tutorial will cover the key architectural components of such a system, for each aspect it will discuss the options available, and their unique advantages / disadvantages. In particular it will provide guidance on selecting an appropriate cellular modem and communications protocol (e.g. CSD, SMS, USSD or GPRS). Next the tutorial will discuss the complexities of communicating with and managing the remote device. Lastly it will conclude by explaining the options for collating the remote data and presenting it back to the user through alerts or a web portal.

Bio:
Ian currently leads the technical team at Eseye, a innovative cellular enabling company. Eseye provides end to end solutions for the machine to machine markets, focussing primarily on cellular platforms, air time and enterprise hosting.
Prior to Eseye, Ian was heavily involved in the creation of ZigBee, founding a company called CompXs and chairing the Network Layer Working Group for the ZigBee Alliance.

Related Information Links:
www.eseye.com

15.45
Tutorial 3.6 –

Tom Cooksey, Software Engineer, Nokia-Qt Software
Using OpenGL ES APIs on Embedded Linux

Public expectations of user interfaces to consumer electronics have changed dramatically in the last 2 years. To accommodate these expectations, SoC manufacturers are adding ever more sophisticated graphics hardware to their products. This hardware is usually exposed through the OpenGL ES and related Khronos APIs. This talk will provide a brief overview of these APIs and how they could be used to develop compelling user interfaces. The focus will be on devices running the Linux operating system, however many of the issues relate to any operating system.
The discussion will continue by discussing problems encountered when trying to use these APIs for tasks such as window compositing, animations, 2D vector graphics and special effects. Issues range from memory management across processes to OpenGL ES only working with triangles. The lessons learned are that good partnerships with hardware manufacturers are important, OpenGL ES is not a good API for vector graphics and that copying memory is expensive! Attendees should leave with a better knowledge of graphics acceleration under Linux and understand that having a hardware accelerator does not automatically produce faster graphics.

Bio:
Tom Cooksey is a software engineer at the Qt Software division of Nokia (formally Trolltech) and works on hardware accelerated graphics in Qt for Embedded Linux's windowing system - QWS. Tom has an MEng in Computing & Communication Systems Engineering from UMIST (now part of The University of Manchester) and is also a member of the IET. Before moving to Oslo to work for Trolltech, he worked at BT's Adastral Park research facility for 2 years and before that was at HP-Labs in Bristol.

Related Information Links:
www.trolltech.com
www.khronos.org

16.45
DRINKS RECEPTION

Thursday, 2nd October

Room:
Gallery 2

Dependable Systems - Code Generation

- Development and Verification Standards in the Embedded Market
- Developing for System and Software Security
- Testing Hardware for Reliability
- Software Development for Critical Systems

9.30 **WELCOME**

09.45
Tutorial 4.1 -

Matthew Fowler, CEO, New Technology/enterprise (NT/e)
Code Generation - The Fundamentals

A quick gallop through the field of code generation for those new to the field, to give you enough background to appreciate later sessions.

This will cover:

- The basics: templates and variables.
- Code generation: building up to systems: coordinated builds.
- The various styles of "code generation".
- Intent: saying what we mean. Models, DSLs, text vs graphical, meta-models.
- Doing the generation. Increasing abstraction; model-to-model transforms.
- What's it like doing it for yourself.
- The industry players.
- What the future looks like.

Bio:
Matthew Fowler is founder and CEO of New Technology/enterprise (NT/e). Matthew received a BSc degree, Computer Science from MIT in 1974. He has created products in many areas of software - starting with system generation in the late 70s, LANs, WANs and software tools in the 80s, and more recently language design and implementation. He has also acted as trainer and consultant on many large-scale applications. His current interest is large-scale system generation.

Related Information Links:

10.45
Tutorial 4.2

Speaker TBA
MDA Explained. (Describing the OMG's MDA approach from an embedded perspective.)

11.45
COFFEE BREAK

12.00
Tutorial 4.3

Karsten Thoms, Software Architect, Itemis
Introduction to the Eclipse Modelling Project

This tutorial gives an overview of complementary and alternative technologies from the Eclipse Modelling world. The Eclipse Modelling Project is the starting point for model-based development technologies in the open source community. It provides a unified set of modelling frameworks, tooling and standards implementation and is rapidly increasing with new subprojects, components and developers. Despite, or because of this, it is hard to get into the larger world of Eclipse Modelling. This tutorial will demonstrate the usage of some of the components that have been proven in real world model-driven projects, including Eclipse Modelling Framework (metamodeling, tooling), Xpand (model-to-text), Xtend (model-to-model), Xtext (textual DSLs) and Check (model validation).

Bio:
Karsten Thoms is a Software Architect at itemis, the leading company for Model Driven Software Development in Germany. As part of his work he helps customers realize MDSD in real-life projects. Karsten is one of the core developers on the openArchitectureWare project and gives training courses for this product regularly.

Related Information Links:

13.00
LUNCH

13.45
Tutorial 4.4 –

Karsten Thoms, Software Architect, Itemis
Model it your way - Efficient modelling using Domain Specific Languages

When people talk about software models they often think of UML models. UML is a General Purpose Modelling Language (GPML) that can be applied to several modelling domains. But this flexibility has some severe drawbacks: The UML has an enormous complexity and yet it is not sufficient to express common software modelling aspects, e.g. describing the GUI of applications.

Domain Specific Languages (DSLs) are languages tailored to a specific problem domain, or often a specific project. Both visual and textual DSLs exist and both have their pros and cons.

What are the benefits of choosing DSLs instead of GPMLs for model-driven software development? Can the use of DSLs conform to the OMG's Model-Driven Architecture paradigm? What are the disadvantages of using UML and associated tools in practice?

It might sound that building your own languages and appropriate tooling is too complex and expensive, but this session will demonstrate that nowadays there is no magic in designing DSLs and providing tooling for them.

Bio:
Karsten Thoms is a Software Architect at itemis, the leading company for Model Driven Software Development in Germany. As part of his work he helps customers realize MDSD in real-life projects. Karsten is one of the core developers on the openArchitectureWare project and gives training courses for this product regularly.

Related Information Links:

14.45
Tutorial 4.5

Axel Terfloth, Itemis
Testing Code Generators

Embedded system software must provide a high degree of reliability, especially in the context of safety-critical systems, so appropriate testing strategies are a necessity. When applying model-driven approaches another category of software is added to the development process. This is the code that processes models and generates code. Also the implementation of code generators has to be reliable.

This tutorial discusses testing strategies for code generators as they are applied in the automotive domain for AUTOSAR-based ECU software. It demonstrates how those testing strategies that include unit testing and coverage metrics measurement can be applied using the Eclipse openArchitectureWare MDSD framework.

15.45
PANEL SESSION

Discussion: 10 Years from now nobody will write code by hand

Moderator: Andrew Watson, Technical Director, OMG
Panellists: Matthew Fowler, Karsten Thoms, others to be confirmed.

Today, knowingly or unknowingly, most application programmers use code generation tools in some form. Whether they're simple macros, C++ templates, Generators in Ruby on Rails, or Java tools like Xdoclet and EJBgen, code generators help programmers write code fragments that fall into predictable patterns. However, some tools aim to go further, and allow almost all an application's code to be automatically created from a concise description in UML or a special-purpose Domain Specific Language. Can tools like JeeWiz!, openArchitectureWare and AndroMDA deliver on this promise to write most of your code for you, or are the benefits of Code Generation over-hyped? Participate in our panel to hear what the experts think and share your views.

16.45
DRINKS RECEPTION

Thursday, 2nd October

Room:
Gallery 3

Hands-On Class
Presented by Feabhas Ltd

09.45 – 12.45

Embedded Systems Concurrency with C++

Workshop Description:
One of the key factors of embedded systems programming is the concurrent nature of the real world and the need to deal with this parallelism. Concurrency handling in embedded systems is typically provided by a Real-Time Operating System (RTOS). These RTOS predominantly feature a C API.

C++ is gradually becoming more prevalent in embedded systems. This workshop focuses on developing a C++- based, object-oriented wrapper for a 'classic' RTOS. This provides an abstraction from the underlying platform, leading to far more portable and maintainable code.

The skills learnt in this workshop can be transferred to most operating systems – including Windows and Linux.

The workshop is 50% practical and 50% theory.

Workshop Objectives:
- To dispel some of the myths of concurrent programming
- To introduce the practical necessities of multi-tasking programming in an embedded systems.

Delegates will learn:
- The features of commercial RTOS
- The issues involved in concurrent programming
- How to write C++ code to provide an object-oriented abstraction of a typical RTOS.

Who Should Attend:.
- Software engineers who are developing embedded systems in C++

Pre-Requisites
- Good C++ programming skills, including:
- General knowledge of an RTOS or embedded operating systems is useful

Duration:
3 hours.

Course Materials:
Student workbook.

Course Workshop:
The workshop practical will be PC-based using the WIN32 API.

Related Information Links:

16.45
DRINKS RECEPTION

Thursday, 2nd October

Room:
Gallery 3

Hands-On Class
Presented by Feabhas Ltd

13.45 – 16.45

Embedded Linux: a practical approach

Workshop Description:
This session shows how to build a working embedded Linux system, using a Power PC development board as the target. Topics covered include cross-development tools, kernel configuration and compilation and the essential components of a root file system. The combination of theory and practice will give delegates an insight into the working of Linux in embedded environments. The workshop is 50% practical and 50% theory.

Workshop Objectives:
- To get real-world exposure to embedded Linux
- To develop an application to run on an embedded Linux system.

Delegates will learn:
- How to configure a standard Linux kernel for use in a cross development system.
- The steps to write, compile and download an embedded Linux application with real hardware.

Who Should Attend:
- Engineers wishing to assess the suitability of Linux for their next application.
- Software engineers who are developing applications for embedded Linux.

Pre-Requisites
- Good 'C' programming skills
- General knowledge of an RTOS or embedded operating systems
- Knowledge of Linux/Unix commands (e.g. ls, cat, cp) will be useful, but not essential

Duration:
3 hours.

Course Materials:
Student workbook.

Course Workshop:
The course presents embedded concepts applied to Linux, using an ARM development board as the target (ConnectCore Wi-9C Digi JumpStart Kit for Linux). The host development system will be a standard PC running desktop Linux. We use the target as an example of a simple embedded system which can control hardware via a simple digital I/O interface.

Related Information Links:

16.45
DRINKS RECEPTION